

Fundamental Study

Semantics of order-sorted specifications

Uwe Waldmann

Max-Planck-Institut für Informatik, Im Stadtwald, W-6600 Saarbrücken, Germany

Communicated by M. Nivat

Received June 1989

Revised September 1990

Abstract

Waldmann, U., Semantics of order-sorted specifications, *Theoretical Computer Science* 94 (1992) 1–35.

Order-sorted specifications (i.e. many-sorted specifications with subsort relations) have been proved to be a useful tool for the description of partially defined functions and error handling in abstract data types.

Several definitions for order-sorted algebras have been proposed. In some papers an operator symbol, which may be multiply declared, is interpreted by a family of functions (“overloaded” algebras). In other papers it is always interpreted by a single function (“non-overloaded” algebras). On the one hand, we try to demonstrate the differences between these two approaches with respect to equality, rewriting and completion; on the other hand, we prove that in fact both theories can be studied in parallel provided that certain notions are suitably defined.

The overloaded approach differs from the many-sorted and the nonoverloaded one in that the overloaded term algebra is not necessarily initial. We give a decidable sufficient criterion for the initiality of the term algebra, which is less restrictive than GJM-regularity as proposed by Goguen, Jouannaud and Meseguer.

Sort-decreasingness is an important property of rewrite systems since it ensures that confluence and Church–Rosser property are equivalent, that the overloaded and nonoverloaded rewrite relations agree, and that variable overlaps do not yield critical pairs. We prove that it is decidable whether or not a rewrite rule is sort-decreasing, even if the signature is not regular.

Finally, we demonstrate that every overloaded completion procedure may also be used in the nonoverloaded world, but not conversely, and that specifications exist that can only be completed using the nonoverloaded semantics.

Contents

1. Introduction.	2
2. Foundations.	3
2.1. Basic notations.	3
2.2. Signatures and algebras.	4
2.3. Variables.	9
2.4. Equations.	10
2.5. Remarks.	15
3. Rewriting.	17
3.1. Rewrite relations.	17
3.2. Sort-decreasingness.	18
3.3. Confluence and Church–Rosser property.	20
3.4. Remarks.	22
4. Completion.	25
4.1. Unification and regularity.	25
4.2. Critical pairs.	27
4.3. A completion procedure.	27
4.4. Overloaded versus nonoverloaded completion.	30
4.5. Remarks.	32
5. Conclusion.	32
References.	33

1. Introduction

In mathematics and computer science, conventionally, every object occurring in some formula has a certain type. A variable n generally represents a natural number, δ ranges over real numbers and the exponential function maps real numbers to real numbers. Often, these domains are related by inclusions, for example, the set of naturals is a subset of the reals, so that in the term $\exp(n)$ the variable n may occur at a position where a real number is expected. Besides, operators are overloaded, so that the symbol $+$ is used for the addition of naturals as well as for the addition of real and complex numbers and even vectors. On the one hand, overloading simplifies the notation; on the other hand, it is useful to express the similarity of these different operations (e.g. they all are associative and commutative).

The use of types, subsorts, and overloading for logic, specification and programming was already proposed by Oberschelp [33]; later it became well-known especially through Goguen et al. [16, 18, 19, 20]. Order-sorted specifications simplify the presentation of partially defined functions and allow a more elegant formulation of error recovery and error propagation in algebraic specifications [14, 15, 39]. Using a logic with subsorts, the efficiency of an automatic theorem prover can be increased [3, 35, 43]. The typing mechanisms of several programming languages are based on order-sorted signatures; apart from the OBJ family [9, 21, 27] we should mention Smolka’s language TEL [38], where a combination of subsorts and polymorphism is used. A three-level system of values, types and partially ordered sorts was introduced by Nipkow and Snelting [32].

Several definitions for order-sorted algebras have been proposed. In some of the above-mentioned papers an operator symbol, which may be multiply declared, is interpreted by a family of functions; in other papers it is always interpreted by a single function (or by a family of functions that may be considered as restrictions of a single function). Usually, the first approach is called “overloaded”; the second one “nonoverloaded” (although it could be argued that in “nonoverloaded” algebras overloading is restricted but still possible).

In general, overloaded and nonoverloaded algebras do not induce the same notion of equality on the set of terms; and the corresponding definitions and theorems for the overloaded and the nonoverloaded case often differ in unobtrusive details since frequently the additional possibilities of overloaded algebras have to be paid for with restrictions.

In this paper we investigate equality, rewriting, and completion in order-sorted signatures, following Gnaedig et al. [13]. In contrast to Poigné’s and Stell’s more categorial approaches [34, 40] we lay stress on the pragmatismal comparison of overloaded and nonoverloaded semantics: on the one hand, we try to demonstrate the differences and pitfalls; on the other hand, we prove that both theories can be studied in parallel provided that certain notions are suitably defined.

2. Foundations

2.1. Basic notations

We use the standard symbols \in , \cap , \cup , \setminus and \times for the membership relation and the set theoretic operations intersection, union, set difference and cartesian product. The subset relation is denoted by \subseteq . The symbol \emptyset represents the empty set, the set of natural numbers (including 0) is abbreviated by \mathbb{N} . The expression A^n denotes the set of all tuples or strings over A with length n , the letter ε symbolizes the empty tuple or empty string. Finally, A^* and A^+ are defined by $A^* := \bigcup_{n \geq 0} A^n$ and $A^+ := \bigcup_{n \geq 1} A^n$.

A function f with domain A and range B is written as $f: A \rightarrow B$; for some $A' \subseteq A$ the set $f(A') := \{f(x) \mid x \in A'\}$ is the image of A' under f and $f|_{A'}$ is the restriction of f to A' . The symbol id_A means the identity function from A to A . Given two functions $f: A \rightarrow B$ and $g: B \rightarrow C$, the composition $(g \circ f): A \rightarrow C$ of f and g is the function that maps every $x \in A$ to $g(f(x)) \in C$; this is denoted by $x \mapsto g(f(x))$.

A binary relation \leq over A that is reflexive and transitive is a quasi-ordering; if it is also antisymmetric, it is called a partial ordering. An antireflexive, transitive and antisymmetric relation is called a strict ordering. Let (A, \leq) be a partially ordered set. An element $a \in A$ is maximal if $a \leq b$ implies $a = b$ for every $b \in A$. We say that a is the greatest element of A if $b \leq a$ holds for every $b \in A$. Analogously, we define “minimal” and “least” elements. A strict ordering $<$ over A is said to be noetherian if there is no infinite sequence (a_0, a_1, \dots) such that $a_{i+1} < a_i$ for all $i \in \mathbb{N}$.

Given two binary relations $\rightarrow_1 \subseteq A \times B$ and $\rightarrow_2 \subseteq B \times C$, we define their composition $(\rightarrow_1 \circ \rightarrow_2) \subseteq A \times C$ by $x(\rightarrow_1 \circ \rightarrow_2)z \Leftrightarrow \exists y \in B: x \rightarrow_1 y, y \rightarrow_2 z$. Let \rightarrow be a binary relation over A . The inverse relation of \rightarrow is abbreviated by \leftarrow and the symmetric closure by \leftrightarrow . The symbol \rightarrow^+ represents the transitive closure, \rightarrow^* is the reflexive and transitive closure and \leftrightarrow^* is the equivalence closure of \rightarrow .

2.2. Signatures and algebras

Definition 2.1. An order-sorted signature is a triple (S, \leq, Σ) , where S is a set of sorts, \leq a partial ordering over S , and Σ a family $\{\Sigma_{w,s} \mid w \in S^*, s \in S\}$ of (not necessarily disjoint) sets of operator symbols.

The equivalence closure $(\leq \cup \geq)^*$ of the relation \leq is denoted by \cong . The equivalence classes of S modulo \cong are called connected components of S . The ordering \leq is extended componentwise to strings $s_1 \dots s_n \in S^*$; so, we have $s_1 \dots s_n \leq s'_1 \dots s'_n$ if and only if $s_i \leq s'_i$ for $1 \leq i \leq n$.

In order to make the notation simpler and more intuitive, we shall often write $f: w \rightarrow s$ instead of $f \in \Sigma_{w,s}$ and $f: \rightarrow s$ instead of $f \in \Sigma_{\epsilon,s}$. We also use Σ as an abbreviation for both (S, \leq, Σ) and $\bigcup_{w,s} \Sigma_{w,s}$.

Definition 2.2. The set $T_{\Sigma,s}$ of ground terms over Σ with sort s is the least set with the following properties:

- (i) $f \in T_{\Sigma,s}$ if $f: \rightarrow s_0$ and $s_0 \leq s$.
- (ii) $f(t_1, \dots, t_n) \in T_{\Sigma,s}$ if $f: s_1 \dots s_n \rightarrow s_0$ such that $s_0 \leq s$ and $t_i \in T_{\Sigma,s_i}$ for every $i \in \{1, \dots, n\}$.

$T_\Sigma := \bigcup_{s \in S} T_{\Sigma,s}$ denotes the set of all ground terms over Σ .

Sometimes we need a more general notion of terms, which does not have the sort constraints of the previous definition.

Definition 2.3. The set ET_Σ of extended ground terms over Σ is the least set with the following properties:

- (i) $f \in ET_\Sigma$ if $f: \rightarrow s_0$.
- (ii) $f(t_1, \dots, t_n) \in ET_\Sigma$ if $f: s_1 \dots s_n \rightarrow s_0$ and $t_i \in ET_\Sigma$ for $1 \leq i \leq n$.

The set T_Σ is a subset of ET_Σ . If an extended term $t \in ET_\Sigma$ is an element of T_Σ , we say that t is a well-formed term; otherwise, t is called ill-formed. As shown by Comon [4, 5] an order-sorted signature can be considered as a finite bottom-up tree automaton. Then ET_Σ is the set of all trees over the alphabet of the automaton, $T_{\Sigma,s}$ is the subset of ET_Σ that is recognized by the automaton in the final state s .

As usual, positions (also known as occurrences) of a term are denoted by strings of natural numbers. The set of all positions of an extended term $t \in \text{ET}_\Sigma$ is $\text{Pos}(t)$; the subterm of t at the position $p \in \text{Pos}(t)$ is written t/p . Given a position $p \in \text{Pos}(t)$, the result of the replacement of the subterm at p in t by t' is written as $t[p \leftarrow t']$. Note that $t[p \leftarrow t']$ may be ill-formed even if t and t' are well-formed terms.

Definition 2.4. The spectrum $\text{spectr}(t)$ of a term $t \in \text{T}_\Sigma$ is the set of all sorts $s \in S$ such that $t \in \text{T}_{\Sigma, s}$. The set of all spectra of terms in T_Σ is denoted by spectr_Σ .

The spectrum of a term can be computed using the following recursion formulae:

$$\text{spectr}(f) = \{s \in S \mid f: \rightarrow \bar{s}, \bar{s} \leq s\}$$

$$\text{spectr}(f(t_1, \dots, t_n)) = \{s \in S \mid f: s_1 \dots s_n \rightarrow \bar{s}, \bar{s} \leq s, \forall i \leq n: s_i \in \text{spectr}(t_i)\}$$

The following lemma is due to Schmidt–Schauß [36].

Lemma 2.5. *If the signature (S, \leq, Σ) is finite, then spectr_Σ is finite and effectively computable. Besides, there exists a finite and computable set $Q \subseteq \text{T}_\Sigma$ such that for every spectrum $M \in \text{spectr}_\Sigma$ there is a term $t \in Q$ satisfying $M = \text{spectr}(t)$.*

Definition 2.6. Let (S, \leq, Σ) be an order-sorted signature. An overloaded (S, \leq, Σ) -algebra A (Σ - $\mathcal{C}\mathcal{V}$ -algebra) consists of a family $\{A_s \mid s \in S\}$ of sets and a function $A_f^{w, s}: A_w \rightarrow A_s$ for every $f \in \Sigma_{w, s}$ such that the following conditions are fulfilled:

- (i) $A_s \subseteq A_{s'}$ if $s \leq s'$.
- (ii) $A_f^{w, s}$ equals $A_f^{w', s'}$ on A_w if $s \leq s'$, $w \leq w'$ and $f \in \Sigma_{w, s} \cap \Sigma_{w', s'}$.

We use A_{s_1, \dots, s_n} as an abbreviation for $A_{s_1} \times \dots \times A_{s_n}$, A_ε is some one-point set. (The functions $A_f^{w, s}$ may be regarded as constants.)

Definition 2.7. Let (S, \leq, Σ) be an order-sorted signature. A nonoverloaded (S, \leq, Σ) -algebra A (Σ - $\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra) consists of a family $\{A_s \mid s \in S\}$ of sets and a function $A_f: D_f^A \rightarrow C_A$ for every $f \in \Sigma$ such that the following conditions are fulfilled:

- (i) $A_s \subseteq A_{s'}$ if $s \leq s'$.
- (ii) D_f^A is a subset of $(C_A)^*$, where $C_A := \bigcup_{s \in S} A_s$.
- (iii) If $f \in \Sigma_{w, s}$, then $A_w \subseteq D_f^A$ and $A_f(A_w) \subseteq A_s$.

A_{s_1, \dots, s_n} and A_ε are defined as in the overloaded case.

Obviously, we can make T_Σ (which we abbreviate as T) a Σ - $\mathcal{C}\mathcal{V}$ -algebra by defining $T_s := \text{T}_{\Sigma, s}$ and $T_f^{w, s}(t_1, \dots, t_n) := f(t_1, \dots, t_n)$ for $f: w \rightarrow s$, $w = s_1 \dots s_n$ and $t_i \in T_{s_i}$. Analogously, we can make T_Σ a Σ - $\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra by defining $T_s := \text{T}_{\Sigma, s}$; in this case, for an operator symbol $f \in \Sigma$ we define $D_f^T := \bigcup_{f: w \rightarrow s} T_w$ and $T_f(t_1, \dots, t_n) := f(t_1, \dots, t_n)$ for $(t_1, \dots, t_n) \in D_f^T$.

Definition 2.8. Let A and B be two (S, \leq, Σ) - \mathcal{CV} -algebras. An (S, \leq, Σ) - \mathcal{CV} -homomorphism $h: A \rightarrow B$ is an S -sorted family of functions $\{h_s: A_s \rightarrow B_s \mid s \in S\}$ such that

- (i) h_s equals $h_{s'}$ on A_s if $s \leq s'$.
- (ii) $h_s(A_f^{w,s}(\alpha_1, \dots, \alpha_n)) = B_f^{w,s}(h_{s_1}(\alpha_1), \dots, h_{s_n}(\alpha_n))$ for all $f \in \Sigma_{w,s}$, $w = s_1 \dots s_n$, and $(\alpha_1, \dots, \alpha_n) \in A_w$.

An \mathcal{CV} -homomorphism $h: A \rightarrow B$ is called an \mathcal{CV} -isomorphism if an \mathcal{CV} -homomorphism $h': B \rightarrow A$ exists satisfying $h' \circ h = \text{id}_A$ and $h \circ h' = \text{id}_B$. Here the composition operator \circ is meant componentwise.

Definition 2.9. Let A and B be two (S, \leq, Σ) - $\mathcal{N}\mathcal{CV}$ -algebras. An (S, \leq, Σ) - $\mathcal{N}\mathcal{CV}$ -homomorphism $h: A \rightarrow B$ is a function $h: C_A \rightarrow C_B$ such that

- (i) $h(A_s) \subseteq B_s$ for each $s \in S$.
- (ii) $h(D_f^A) \subseteq D_f^B$ and $h(A_f(\alpha_1, \dots, \alpha_n)) = B_f(h(\alpha_1), \dots, h(\alpha_n))$ for all $f \in \Sigma$ and $(\alpha_1, \dots, \alpha_n) \in D_f^A$.

An $\mathcal{N}\mathcal{CV}$ -homomorphism $h: A \rightarrow B$ is called an $\mathcal{N}\mathcal{CV}$ -isomorphism if an $\mathcal{N}\mathcal{CV}$ -homomorphism $h': B \rightarrow A$ exists satisfying $h' \circ h = \text{id}_A$ and $h \circ h' = \text{id}_B$.

For every fixed signature (S, \leq, Σ) the Σ - \mathcal{CV} -algebras and Σ - \mathcal{CV} -homomorphisms make up the category $\text{OSA}_S^{\mathcal{CV}}$, and the Σ - $\mathcal{N}\mathcal{CV}$ -algebras and Σ - $\mathcal{N}\mathcal{CV}$ -homomorphisms make up the category $\text{OSA}_S^{\mathcal{N}\mathcal{CV}}$.

Using overloaded homomorphisms as described above it can happen that an equation is satisfied by some algebra A and is not satisfied by some other algebra A' isomorphic to A [20, 31]. The subsequent condition excludes such a situation. (For the same reason later we will have to restrict ourselves to equations $t \approx t'$, where t and t' are members of the same connected component of the sort set S .)

Definition 2.10. A signature (S, \leq, Σ) is called coherent if, whenever two sorts s, s' are contained in the same connected component of S , then there is a sort s° such that $s \leq s^\circ$ and $s' \leq s^\circ$.

The claim for coherence constitutes a considerable restriction of the notion of homomorphisms. If (S, \leq, Σ) is coherent, every \mathcal{CV} -homomorphism can be described as an (S/\cong) -sorted family of functions; if S consists of only one connected component, it can even be regarded as a single function.

Nonoverloaded algebras and homomorphisms do not cause such problems; thus, coherence is unnecessary here.

Definition 2.11. A Σ - \mathcal{CV} -algebra A is called initial in the set of all overloaded Σ -algebras if for every Σ - \mathcal{CV} -algebra B there is exactly one \mathcal{CV} -homomorphism $h: A \rightarrow B$ (analogously, for $\mathcal{N}\mathcal{CV}$ -algebras).

Definition 2.12. Let (S, \leq, Σ) be a signature and A be a Σ - \mathcal{CV} -algebra. An element $\alpha \in A$ is called an \mathcal{CV} -interpretation of $t \in T_S$ if

- (i) $t = f(t_1, \dots, t_n)$ for some $n \geq 0$,

- (ii) $f: s_1 \dots s_n \rightarrow s$,
- (iii) $t_i \in \mathbf{T}_{\Sigma, s_i}$ for all $i \in \{1, \dots, n\}$,
- (iv) $\alpha_i \in A_{s_i}$ is an interpretation of t_i for all $i \in \{1, \dots, n\}$,
- (v) $\alpha = A_f^{s_1 \dots s_n, s}(\alpha_1, \dots, \alpha_n)$.

Obviously, every term $t \in \mathbf{T}_{\Sigma}$ has at least one interpretation in every Σ - \mathcal{CV} -algebra. However, it can happen that a term has more than one interpretation in certain algebras.

Definition 2.13. A signature (S, \leq, Σ) is called \mathcal{CV} -consistent if every term $t \in \mathbf{T}_{\Sigma}$ has exactly one interpretation in every Σ - \mathcal{CV} -algebra.

If condition (v) in Definition 2.12 is replaced by $\alpha = A_f(\alpha_1, \dots, \alpha_n)$, we can similarly define the $\mathcal{N}\mathcal{CV}$ -interpretation of a term. However, as the $\mathcal{N}\mathcal{CV}$ -interpretation is always uniquely determined, every signature is trivially $\mathcal{N}\mathcal{CV}$ -consistent. Therefore “consistent” will subsequently always mean “ \mathcal{CV} -consistent” unless explicitly said otherwise.

Theorem 2.14. Given a signature (S, \leq, Σ) , the following three properties are equivalent:

- (i) (S, \leq, Σ) is consistent.
- (ii) For every $n \in \mathbb{N}$ and every $t = f(t_1, \dots, t_n) \in \mathbf{T}_{\Sigma}$ the set $L(t) := \{ws \in S^+ \mid w = s_1 \dots s_n, f: w \rightarrow s, t_i \in \mathbf{T}_{\Sigma, s_i}\}$ has exactly one equivalence class modulo the relation $\cong_{L(t)}$, where $\cong_{L(t)}$ denotes the equivalence closure of $\leq \cap (L(t) \times L(t))$.
- (iii) For every $n \in \mathbb{N}$, every operator $f \in \Sigma$, and for all spectra $M_1, \dots, M_n \in \text{spctr}_{\Sigma}$ the set $L := \{ws \in S^+ \mid w = s_1 \dots s_n, f: w \rightarrow s, s_i \in M_i\}$ is empty or has exactly one equivalence class modulo the relation \cong_L .

Proof. We show at first that (i) implies (ii). For $t = f(t_1, \dots, t_n) \in \mathbf{T}_{\Sigma}$ and $ws \in L(t)$ let $[ws]_t := \{\tilde{w}\tilde{s} \in L(t) \mid \tilde{w}\tilde{s} \cong_{L(t)} ws\}$ be the equivalence class of ws modulo $\cong_{L(t)}$. Now we simultaneously define a Σ - \mathcal{CV} -algebra I and a function $\text{term}: I \rightarrow \mathbf{T}_{\Sigma}$ such that $\text{term}(I_s) \subseteq \mathbf{T}_{\Sigma, s}$ for all $s \in S$.

- $\langle f, [\tilde{s}]_f \rangle \in I_s$ if $f: \rightarrow \tilde{s}$ and $\tilde{s} \leq s$.

$$\text{term}(\langle f, [\tilde{s}]_f \rangle) := f.$$

- $\langle f, [s_1 \dots s_n \tilde{s}]_{f(t_1, \dots, t_n)} \rangle(\alpha_1, \dots, \alpha_n) \in I_s$ if $f: s_1 \dots s_n \rightarrow \tilde{s}$, $\tilde{s} \leq s$, and if $\alpha_i \in I_{s_i}$ and $t_i = \text{term}(\alpha_i)$ hold for every $i \in \{1, \dots, n\}$.

$$\text{term}(\langle f, [s_1 \dots s_n \tilde{s}]_{f(t_1, \dots, t_n)} \rangle(\alpha_1, \dots, \alpha_n)) := f(\text{term}(\alpha_1), \dots, \text{term}(\alpha_n)).$$

- Nothing is in I unless it so follows from the preceding rules.

For $f: s_1 \dots s_n \rightarrow s$ with $n \geq 0$ let $I_f^{s_1 \dots s_n, s}$ be defined by

$$I_f^{s_1 \dots s_n, s} : I_{s_1} \times \dots \times I_{s_n} \rightarrow I_s,$$

$$(\alpha_1, \dots, \alpha_n) \mapsto \langle f, [s_1 \dots s_n \tilde{s}]_{f(t_1, \dots, t_n)} \rangle(\alpha_1, \dots, \alpha_n), \quad \text{where } t_i := \text{term}(\alpha_i).$$

It is easily proved that I is a $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebra and that $\text{term}(x)=t$ holds if $x \in I_s$ is an interpretation of $t \in T_{\Sigma, s}$.

Suppose that (S, \leq, Σ) is consistent and $f(t_1, \dots, t_n) \in T_{\Sigma}$ for some $n \geq 0$. Assume that ws and $w's'$ are contained in $L(t)$. For $i \in \{1, \dots, n\}$ let α_i be the uniquely determined interpretation of t_i . As $\text{term}(\alpha_i)=t_i$ holds, we know that $I_f^{w, s}(\alpha_1, \dots, \alpha_n) = \langle f, [ws]_{f(t_1, \dots, t_n)} \rangle(\alpha_1, \dots, \alpha_n)$ and $I_f^{w', s'}(\alpha_1, \dots, \alpha_n) = \langle f, [w's']_{f(t_1, \dots, t_n)} \rangle(\alpha_1, \dots, \alpha_n)$; since both values are interpretations of t , we have $ws \cong_{L(t)} w's'$.

We now prove the (ii) \Rightarrow (i) part by induction on the term structure. Let A be a $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebra and let $t = f(t_1, \dots, t_n) \in T_{\Sigma}$, where $n \geq 0$. By the induction hypothesis t_i has a uniquely determined interpretation α_i in A for all $i \in \{1, \dots, n\}$. Suppose that $\alpha = A_f^{w, s}(\alpha_1, \dots, \alpha_n)$ and $\alpha' = A_f^{w', s'}(\alpha_1, \dots, \alpha_n)$ are interpretations of t . Since $ws, w's' \in L(t)$ and since $L(t)$ has exactly one equivalence class modulo the relation $\cong_{L(t)}$, there is a sequence $w^0 s^0, \dots, w^m s^m$ of elements of $L(t)$ such that $ws = w^0 s^0$, $w's' = w^m s^m$ and $w^{k-1} s^{k-1} \leq w^k s^k$ or $w^{k-1} s^{k-1} \geq w^k s^k$ holds for every $k \in \{1, \dots, m\}$. As $(\alpha_1, \dots, \alpha_n)$ is a member of A_{w^k} for each $k \in \{0, \dots, m\}$, we have $A_f^{w^{k-1}, s^{k-1}}(\alpha_1, \dots, \alpha_n) = A_f^{w^k, s^k}(\alpha_1, \dots, \alpha_n)$; this implies $\alpha = A_f^{w', s'}(\alpha_1, \dots, \alpha_n) = \alpha'$.

Because for every term $t_i \in T_{\Sigma}$ there is a spectrum $M_i \in \text{spctr}_{\Sigma}$ (and conversely) such that $M_i = \text{spctr}(t_i)$, the equivalence of (ii) and (iii) is obvious. \square

From condition (iii) we can conclude that the consistency of a finite signature is decidable.

Theorem 2.15. *For every coherent signature Σ the following properties are equivalent:*

- (i) Σ is consistent.
- (ii) *The overloaded term algebra T_{Σ} is initial in the class of all $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebras, and for all terms t such that $t \in T_{\Sigma, s} \cap T_{\Sigma, s'}$, we have $s \cong s'$.*

Proof. Let Σ be a consistent signature and A be an arbitrary $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebra. The function $i: T_{\Sigma} \rightarrow A$ maps every term $t \in T_{\Sigma}$ to its interpretation in A . Defining $i_s := i|_{T_{\Sigma, s}}$, we can show that i is the unique Σ -homomorphism from T_{Σ} to A by induction on the term structure; thus, the algebra T_{Σ} is initial.

Next consider the final $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebra C : For $s \in S$ we have $C_s := \{[s]\}$, where $[s]$ denotes the equivalence class $\{s' \in S \mid s' \cong s\}$ of s modulo \cong ; furthermore, $C_f^{w, s}$ is the constant function mapping every tuple $(c_1, \dots, c_n) \in C_w$ to $[s]$. It is easy to check that $[s]$ is an interpretation of t in C if $t \in T_{\Sigma, s}$ holds. Suppose that $t \in T_{\Sigma, s} \cap T_{\Sigma, s'}$, then $[s]$ and $[s']$ are interpretations of t ; by consistency we have $[s] = [s']$ and, thus, $s \cong s'$.

To prove the reverse direction let A be a $\Sigma\text{-}\mathcal{C}\mathcal{Y}$ -algebra and let t be a term that is contained in $T_{\Sigma, s} \cap T_{\Sigma, s'}$. We have $s \cong s'$; since Σ is coherent, there must be some sort s° such that $s \leq s^\circ$ and $s' \leq s^\circ$. It follows that $h_s(t) = h_{s'}(t) = h_{s^\circ}(t)$ for the uniquely determined homomorphism $h: T_{\Sigma} \rightarrow A$. Hence, $h_s(t)$ is independent of s and we can define a function h such that $h_s(t) = h(t)$ for all sorts $s \in \text{spctr}(t)$. Now a simple proof by induction shows that we have for every term $t \in T_{\Sigma}$: If α is an interpretation of t , then $\alpha = h(t)$. Hence, the interpretation of t is uniquely determined. \square

If a term $t \in T_\Sigma$ has two sorts s_1 and s_2 , where $s_1 \not\leq s_2$, the signature Σ is not $\mathcal{C}\mathcal{V}$ -consistent, but the term algebra T_Σ may be initial nevertheless. In this case the homomorphism h from T_Σ to some algebra A is uniquely determined, but the image of t under h may be not uniquely determined (i.e. we can have $h_s(t) \neq h_{s'}(t)$ for $s, s' \in \text{spctr}(t)$); hence, we have to exclude such signatures.

If Σ is consistent and coherent and if h is a homomorphism from T_Σ to some Σ - $\mathcal{C}\mathcal{V}$ -algebra A , then $h_s(t)$ is independent of s (for all sorts $s \in \text{spctr}(t)$). In this situation we shall often omit the sort s and simply write $h(t)$. However, this is in general not possible for homomorphisms $h: A \rightarrow B$, where $A \neq T_\Sigma$.

In the nonoverloaded case coherence is not necessary; besides, every signature is trivially $\mathcal{A}\mathcal{C}\mathcal{V}$ -consistent. Thus, we have the following theorem.

Theorem 2.16. *The nonoverloaded term algebra T_Σ is the initial Σ - $\mathcal{A}\mathcal{C}\mathcal{V}$ -algebra; it is determined uniquely (up to isomorphism).*

2.3. Variables

Definition 2.17. An S -sorted variable set is a family $V = \{V_s \mid s \in S\}$ of disjoint sets.

A variable x of sort s is written as $x:s$. We shall use V as an abbreviation for $\bigcup_{s \in S} V_s$.

Let (S, \leq, Σ) be an order-sorted signature and V be a variable set disjoint from Σ . By componentwise union of Σ and V we get a new signature $(S, \leq, \Sigma \cup V)$ defined by $(\Sigma \cup V)_{\varepsilon, s} := \Sigma_{\varepsilon, s} \cup V_s$ and $(\Sigma \cup V)_{w, s} := \Sigma_{w, s}$ for $w \neq \varepsilon$. Now we can regard the term algebra $T_{\Sigma \cup V}$ as an overloaded or nonoverloaded (S, \leq, Σ) -algebra; this is denoted by $T_\Sigma(V)$.

The set of all variables in a term $t \in T_\Sigma(V)$ is abbreviated by $\text{Var}(t)$.

Definition 2.18. An assignment ν from a variable set V into a Σ -algebra A is a family of functions $\{v_s: V_s \rightarrow A_s \mid s \in S\}$.

Using the overloaded semantics we encounter an additional problem at this point. As demonstrated by the following example, adding variables to a signature Σ may destroy the consistency of Σ .

Example 2.19. The following signature is consistent:

$$(S, \leq) = \{s_3 \leq s_2, s_2 \leq s_1 \leq s_0, s_2 \leq s'_1 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_3, f: s_1 \rightarrow s_0, f: s'_1 \rightarrow s_0, f: s_3 \rightarrow s_0\}.$$

If we add a variable set V such that $(x: s_2) \in V$, however, the term $f(x: s_2)$ may have more than one interpretation in a $(\Sigma \cup V)$ - $\mathcal{C}\mathcal{V}$ -algebra and $T_{\Sigma \cup V}$ is not initial in the set of all $(\Sigma \cup V)$ - $\mathcal{C}\mathcal{V}$ -algebras.

Thus, we must explicitly claim that $(\Sigma \cup X)$ is consistent for every variable set X . A signature Σ having this property is called strongly consistent. The following lemma shows that the strong consistency of finite signatures is decidable.

Lemma 2.20. *Let Y be a variable set containing exactly one variable $y:s$ of every sort $s \in \mathcal{S}$. Then the signature $(\Sigma \cup X)$ is consistent for every variable set X if and only if $(\Sigma \cup Y)$ is consistent.*

Proof. By Theorem 2.14 it is sufficient to show that $\text{spctr}_{\Sigma \cup X} \subseteq \text{spctr}_{\Sigma \cup Y}$ holds for every variable set X . Given a spectrum $M \in \text{spctr}_{\Sigma \cup X}$, let t be a term in $T_{\Sigma \cup X}$ so that $M = \text{spctr}(t)$. We replace every variable in t by a variable from Y with the same sort; this yields a term $t' \in T_{\Sigma \cup Y}$. Obviously, $\text{spctr}(t) = \text{spctr}(t')$; since $\text{spctr}(t') \in \text{spctr}_{\Sigma \cup Y}$, this implies $\text{spctr}_{\Sigma \cup X} \subseteq \text{spctr}_{\Sigma \cup Y}$. The proof of the reverse direction is trivial. \square

Coherence and strong consistency imply that the term algebra is free in the set of all overloaded Σ -algebras; this is proved in exactly the same way as in the unsorted case [22].

Theorem 2.21. *Let Σ be a coherent and strongly consistent signature. Then the $\mathcal{C}\mathcal{F}$ -algebra $T_{\Sigma}(V)$ is the free Σ - $\mathcal{C}\mathcal{F}$ -algebra generated by V , i.e. for every Σ - $\mathcal{C}\mathcal{F}$ -algebra A and every assignment v from V to A there is exactly one $\mathcal{C}\mathcal{F}$ -homomorphism $v^*: T_{\Sigma}(V) \rightarrow A$ that extends v .*

In the nonoverloaded case additional prerequisites are unnecessary.

Theorem 2.22. *The $\mathcal{N}\mathcal{C}\mathcal{F}$ -algebra $T_{\Sigma}(V)$ is the free Σ - $\mathcal{N}\mathcal{C}\mathcal{F}$ -algebra generated by V , i.e. for every Σ - $\mathcal{N}\mathcal{C}\mathcal{F}$ -algebra A and every assignment v from V to A there is exactly one $\mathcal{N}\mathcal{C}\mathcal{F}$ -homomorphism $v^*: T_{\Sigma}(V) \rightarrow A$ that extends v .*

Definition 2.23. A substitution σ is an assignment from a variable set Y into the term algebra $T_{\Sigma}(X)$. In general, the uniquely determined extension $\sigma^*: T_{\Sigma}(Y) \rightarrow T_{\Sigma}(X)$ of σ will also be denoted by σ .

A substitution $\sigma: \{x_1, \dots, x_n\} \rightarrow T_{\Sigma}(X)$ that maps the variables x_1, \dots, x_n to the terms t_1, \dots, t_n , respectively, is written as $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$.

Definition 2.24. A substitution $\sigma: X \rightarrow T_{\Sigma}(Y)$ is called a specialization if it is injective and if it maps every variable $x:s$ from X to a variable (of the same or of a smaller sort).

2.4. Equations

From now on we consider only coherent and strongly consistent signatures in the overloaded case.

Definition 2.25. A Σ -equation is a triple (X, t, t') , where X is a variable set and t and t' are terms from $T_{\Sigma}(X)$. Besides, in the overloaded case we have to claim that t and t' belong to the same connected component of S . (Note that by coherence this condition is satisfied if and only if t and t' have a common sort $s \in S$.)

We usually write $(\forall X)t \approx t'$ instead of (X, t, t') ; if $X = \text{Var}(t) \cup \text{Var}(t')$, we may omit the variable set.

Definition 2.26. A pair (Σ, E) , where Σ is a signature and E is a set of Σ -equations is called a specification.

Definition 2.27. Let $(\forall X)t \approx t'$ be an equation, where $t, t' \in T_{\Sigma}(X)_s$, and let A be a Σ - $\mathcal{C}\mathcal{V}$ -algebra. If $v_s^*(t) = v_s^*(t')$ holds for every assignment $v: X \rightarrow A$, we say that A satisfies the equation $(\forall X)t \approx t'$, which is abbreviated by $A \models_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$.

Let $(\forall X)t \approx t'$ be an equation and A be a Σ - $\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra. If $v^*(t) = v^*(t')$ holds for every assignment $v: X \rightarrow A$, we say that A satisfies the equation $(\forall X)t \approx t'$, which is abbreviated by $A \models_{\mathcal{N}\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$.

A Σ -algebra A satisfies a set E of equations if it satisfies every equation from E ; such an algebra A is called a (Σ, E) -algebra. Provided that every (Σ, E) - $\mathcal{C}\mathcal{V}$ -algebra satisfies the equation $(\forall X)t \approx t'$, we write $E \models_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$; analogously, we write $E \models_{\mathcal{N}\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$ if every (Σ, E) - $\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra satisfies the equation $(\forall X)t \approx t'$.

Whether an equation $(\forall X)t \approx t'$ is satisfied by an algebra or not may depend on the variable set X , as demonstrated in [8, 17]. Obviously, it is always possible to rename the variables in an equation. The following lemma yields a criterion under which circumstances it is even possible to add or to delete a variable.

Lemma 2.28. *Let Σ be a signature and $(\forall X)l \approx r$ be a Σ -equation. If there is a ground term $t \in T_{\Sigma, s}$, or if X contains a variable of a sort $s' \leq s$, then for every Σ -algebra A*

$$A \models (\forall(X \cup \{y:s\}))l \approx r \Leftrightarrow A \models (\forall X)l \approx r.$$

Proof. Without loss of generality, we may assume that $y \notin X$. Suppose that A does not satisfy the equation $(\forall X)l \approx r$. Then there must be an assignment v from X to A such that $v(l) \neq v(r)$. Now we construct an assignment μ from $(X \cup \{y:s\})$ to A . For every $x \in X$ let $\mu(x) := v(x)$. If there is a ground term $t \in T_{\Sigma, s}$, we set $\mu(y) := \alpha$, where α is the interpretation of t in A ; otherwise, X contains a variable $y':s'$, $s' \leq s$; in this case we define $\mu(y) := v(y')$. As $\mu|_X = v$, we have $\mu(l) = v(l) \neq v(r) = \mu(r)$.

To prove the reverse direction assume that the algebra A does not satisfy the equation $(\forall(X \cup \{y:s\}))l \approx r$, i.e. that there is some assignment μ from $X \cup \{y:s\}$ to A such that $\mu(l) \neq \mu(r)$. We define the assignment $v: X \rightarrow A$ by $v := \mu|_X$, and have $v(l) \neq v(r)$; therefore, A does not satisfy the equation $(\forall X)l \approx r$. \square

We have shown that both the initial $\Sigma\text{-}\mathcal{C}\mathcal{V}$ -algebra and the initial $\Sigma\text{-}\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra have the set of all ground terms as their carriers. If equations are absent, the overloaded and the nonoverloaded algebras induce the same notion of equality on the set of terms:

$$\emptyset \models_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t' \Leftrightarrow \emptyset \models_{\mathcal{N}\mathcal{C}\mathcal{V}} (\forall X)t \approx t' \Leftrightarrow t = t'.$$

We encounter quite a different situation if the set of equations is not empty, as shown by the following example (cf. [10]).

Example 2.29. Let Σ and E be defined by

$$(\mathcal{S}, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_0, b: \rightarrow s_1, c: \rightarrow s_2, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0\},$$

$$E = \{a \approx b, a \approx c\}.$$

Consider the following $(\Sigma, E)\text{-}\mathcal{C}\mathcal{V}$ -algebra A :

$$A_{s_0} := \{\alpha, \alpha_1, \alpha_2\}, \quad A_{s_1} := \{\alpha\}, \quad A_{s_2} := \{\alpha\},$$

$$A_u^{\varepsilon, s_0} := \alpha, \quad A_b^{\varepsilon, s_1} := \alpha, \quad A_c^{\varepsilon, s_2} := \alpha, \quad A_f^{s_1, s_0}: \alpha \mapsto \alpha_1, \quad A_f^{s_2, s_0}: \alpha \mapsto \alpha_2.$$

As the operator declarations $f: s_1 \rightarrow s_0$ and $f: s_2 \rightarrow s_0$ are interpreted by different functions in A , the algebra A satisfies the equations $(\forall \emptyset)a \approx b$ and $(\forall \emptyset)a \approx c$, but not $(\forall \emptyset)f(b) \approx f(c)$. In an $\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra every operator symbol corresponds to exactly one function; so, every $(\Sigma, E)\text{-}\mathcal{N}\mathcal{C}\mathcal{V}$ -algebra satisfies the equation $(\forall \emptyset)f(b) \approx f(c)$. Thus, we have $E \models_{\mathcal{N}\mathcal{C}\mathcal{V}} (\forall X)f(b) \approx f(c)$, but not $E \models_{\mathcal{C}\mathcal{V}} (\forall X)f(b) \approx f(c)$.

The different notions of equality are reflected by different inference systems for the overloaded and the nonoverloaded case.

Inference system 2.30. Let Σ be a signature and E be a set of Σ -equations. Using the following rules new equations can be derived from E .

- | | |
|---|---|
| (E1) Reflexivity: | $(\forall X)t \approx t.$ |
| (E2) Symmetry: | $\frac{(\forall X)t \approx t'}{(\forall X)t' \approx t}.$ |
| (E3) Transitivity: | $\frac{(\forall X)t \approx t', (\forall X)t' \approx t''}{(\forall X)t \approx t''}.$ |
| (E4a) $\mathcal{C}\mathcal{V}$ -Congruence: | $\frac{(\forall X)t_1 \approx t'_1, \dots, (\forall X)t_n \approx t'_n}{(\forall X)f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n)}$
if $t_i, t'_i \in T_\Sigma(X)_{s_i}$ and $f: s_1 \dots s_n \rightarrow s_0$. |

- (E4b) $\mathcal{N}\mathcal{C}\mathcal{V}$ -Congruence:
$$\frac{(\forall X)t_1 \approx t'_1, \dots, (\forall X)t_n \approx t'_n}{(\forall X)f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n)}$$
 if both $f(t_1, \dots, t_n)$ and $f(t'_1, \dots, t'_n)$ are well-formed terms (i.e. if $t_i \in \mathsf{T}_{\Sigma}(X)_{s_i}, f: s_1 \dots s_n \rightarrow s_0$ and if $t'_i \in \mathsf{T}_{\Sigma}(X)_{s'_i}, f: s'_1 \dots s'_n \rightarrow s'_0$).
- (E5) Substitutivity: $(\forall X)\sigma t \approx \sigma t'$ if $(\forall Y)t \approx t'$ is an element of the initial set of equations E and if $\sigma: Y \rightarrow \mathsf{T}_{\Sigma}(X)$ is a substitution.

If an equation $(\forall X)t \approx t'$ can be derived from E using the rules (E1)–(E3), (E4a) and (E5), we write $E \vdash_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$ or $t \stackrel{X}{\underset{E}{\equiv}} t'$; if it can be derived using (E1)–(E3), (E4b) and (E5), this is denoted by $E \vdash_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$ or $t \stackrel{X}{\underset{E}{\equiv}} t'$.

Given Σ, X and E , we define a $\Sigma\text{-}\mathcal{C}\mathcal{V}$ -algebra $\mathsf{T}_{\Sigma, E}(X)$ (abbreviated by T) as follows:

$$T_s := \{[t] \mid t \in \mathsf{T}_{\Sigma}(X)_s\},$$

where $[t] := \{t' \in \mathsf{T}_{\Sigma}(X) \mid E \vdash_{\mathcal{C}\mathcal{V}} t \approx t'\}$ denotes the equivalence class of t . For $f: w \rightarrow s, w = s_1 \dots s_n$ let the function $T_f^{w, s}$ be defined by

$$T_f^{w, s}: T_{s_1} \times \dots \times T_{s_n} \rightarrow T_s$$

$$([t_1], \dots, [t_n]) \mapsto [f(t_1, \dots, t_n)]$$

for representatives t_1, \dots, t_n such that $t_i \in \mathsf{T}_{\Sigma}(X)_{s_i}$.

It is easy to verify that $\mathsf{T}_{\Sigma, E}(X)$ is in fact a $\Sigma\text{-}\mathcal{C}\mathcal{V}$ -algebra.

Theorem 2.31. *Let Σ be a signature and E be a set of Σ -equations. Then the following properties are equivalent:*

- (i) $E \vdash_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$.
- (ii) Every $(\Sigma, E)\text{-}\mathcal{C}\mathcal{V}$ -algebra A satisfies the equation $(\forall X)t \approx t'$.
- (iii) The $\mathcal{C}\mathcal{V}$ -algebra $\mathsf{T}_{\Sigma, E}(X)$ satisfies the equation $(\forall X)t \approx t'$.

Proof. The (i) \Rightarrow (ii) part is proved by induction on the length k of the derivation of the equation $(\forall X)t \approx t'$.

To prove that (ii) implies (iii) it is obviously sufficient to show that $\mathsf{T}_{\Sigma, E}(X)$ is a (Σ, E) -algebra. Let $(\forall Y)t \approx t'$ be an equation in E and let v be any assignment from Y to $\mathsf{T}_{\Sigma, E}(X)$. We choose some substitution $\sigma: Y \rightarrow \mathsf{T}_{\Sigma}(X)$ satisfying $\sigma y \in v(y)$ for all $y \in Y$. Since $(\forall Y)t \approx t' \in E$, according to rule (E5) the equation $(\forall X)\sigma t \approx \sigma t'$ is derivable; so, by the definition of $\mathsf{T}_{\Sigma, E}(X)$ we know that $v(t) = [\sigma t] = [\sigma t'] = v(t')$.

To prove the direction from (iii) to (i) assume that $T_{\Sigma, E}(X)$ satisfies some equation $(\forall X)t \approx t'$, i.e. $v^*(t) = v^*(t')$ holds for every assignment v from X to $T_{\Sigma, E}(X)$. We choose $v(x) := [x]$ for $x \in X$ and get $[t] = v^*(t) = v^*(t') = [t']$; so, by the definition of $T_{\Sigma, E}(X)$ the equation $(\forall X)t \approx t'$ must be derivable. \square

The following theorem can now be proved as in the many-sorted case (e.g. as in [8]).

Theorem 2.32. *The $\mathcal{C}\mathcal{V}$ -algebra $T_{\Sigma, E}(X)$ is the free (Σ, E) - $\mathcal{C}\mathcal{V}$ -algebra generated by X , i.e. for every (Σ, E) - $\mathcal{C}\mathcal{V}$ -algebra A and every assignment $v: X \rightarrow A$ there is exactly one $\mathcal{C}\mathcal{V}$ -homomorphism $\tilde{v}: T_{\Sigma, E}(X) \rightarrow A$ such that $v(x) = \tilde{v}_s([x])$ for all $x \in X_s$.*

As a special case of the preceding theorem for $X = \emptyset$, we have the following corollary.

Corollary 2.33. *The $\mathcal{C}\mathcal{V}$ -algebra $T_{\Sigma, E} := T_{\Sigma, E}(\emptyset)$ is initial in the set of all overloaded (Σ, E) -algebras, i.e. for every (Σ, E) - $\mathcal{C}\mathcal{V}$ -algebra A there is exactly one homomorphism $h: T_{\Sigma, E} \rightarrow A$.*

We shall now present the corresponding definitions and theorems for the nonoverloaded case [39]. In order not to make the notations more clumsy by using still more indices we sometimes use the same symbols as in the overloaded case.

Definition 2.34. Let Σ be a signature, X a variable set and E a set of Σ -equations. The Σ - $\mathcal{A}\mathcal{C}\mathcal{V}$ -algebra $T_{\Sigma, E}(X)$ (abbreviated by T) is defined as follows:

$$T_s := \{[t] \mid t \in T_{\Sigma}(X)_s\},$$

where $[t] := \{t' \in T_{\Sigma}(X) \mid E \vdash_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'\}$ denotes the equivalence class of t .

For $f \in \Sigma$ let D_f^T and T_f be defined by

$$D_f^T := \bigcup_{f: w \rightarrow s} T_w$$

and

$$T_f: D_f^T \rightarrow C_T$$

$$([t_1], \dots, [t_n]) \mapsto [f(t_1, \dots, t_n)] \quad \text{for representatives } t_1, \dots, t_n \text{ such that}$$

$$t_i \in T_{\Sigma}(X)_{s_i} \text{ and } f: s_1 \dots s_n \rightarrow s.$$

Theorem 2.35. *Let Σ be a signature and E be a set of Σ -equations. Then the following properties are equivalent:*

- (i) $E \vdash_{\mathcal{C}\mathcal{V}} (\forall X)t \approx t'$.
- (ii) Every (Σ, E) - $\mathcal{A}\mathcal{C}\mathcal{V}$ -algebra A satisfies the equation $(\forall X)t \approx t'$.
- (iii) The $\mathcal{A}\mathcal{C}\mathcal{V}$ -algebra $T_{\Sigma, E}(X)$ satisfies the equation $(\forall X)t \approx t'$.

Theorem 2.36. *The \mathcal{NCOV} -algebra $T_{\Sigma, E}(X)$ is the free (Σ, E) - \mathcal{NCOV} -algebra generated by X , i.e. for every (Σ, E) - \mathcal{NCOV} -algebra A and every assignment $v: X \rightarrow A$ there is exactly one \mathcal{NCOV} -homomorphism $\tilde{v}: T_{\Sigma, E}(X) \rightarrow A$ such that $v(x) = \tilde{v}([x])$ for all $x \in X_s$.*

Corollary 2.37. *The \mathcal{NCOV} -algebra $T_{\Sigma, E} := T_{\Sigma, E}(\emptyset)$ is initial in the set of all nonoverloaded (Σ, E) -algebras.*

Note that it is in general undecidable whether the overloaded and the nonoverloaded semantics of a set of equations coincide.

Theorem 2.38. *Given an order-sorted specification (Σ, E) it is undecidable whether the relations $\frac{\cdot}{\mathcal{C}\mathcal{Y}} \frac{X}{E}$ and $\frac{\cdot}{\mathcal{A}\mathcal{C}\mathcal{Y}} \frac{X}{E}$ agree.*

Proof. Consider a specification (Σ_1, E_1) over a sort set $\{s_1\}$ and two arbitrary ground terms $t, t' \in T_{\Sigma_1}$. The problem to determine whether t and t' are E_1 -equal is in general undecidable [8]. We define a new specification (S, \leq, Σ, E) as follows:

$$(S, \leq) = \{s_1 \leq s_0, s_2 \leq s_0, s_3 \leq s_0\},$$

$$\Sigma = \Sigma_1 \cup \{a: \rightarrow s_2, b: \rightarrow s_3, f: s_2 \rightarrow s_0, f: s_3 \rightarrow s_0\},$$

$$E = E_1 \cup \{a \approx b, t \approx f(a), t' \approx f(b)\}.$$

If $(\forall \emptyset)t \approx t'$ does not follow from E_1 , we have $E \not\vdash_{\mathcal{C}\mathcal{Y}} (\forall \emptyset)t \approx t'$, but $E \vdash_{\mathcal{A}\mathcal{C}\mathcal{Y}} (\forall \emptyset)t \approx t'$; thus, $\frac{\cdot}{\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E} \neq \frac{\cdot}{\mathcal{A}\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$. On the other hand, if t and t' are E_1 -equal, then $\frac{\cdot}{\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$ and $\frac{\cdot}{\mathcal{A}\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$ are the same. Since $E_1 \vdash (\forall \emptyset)t \approx t' \Leftrightarrow \frac{\cdot}{\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E} = \frac{\cdot}{\mathcal{A}\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$, the equality of $\frac{\cdot}{\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$ and $\frac{\cdot}{\mathcal{A}\mathcal{C}\mathcal{Y}} \frac{\emptyset}{E}$ is undecidable. \square

2.5. Remarks

A number of more or less different descriptions of order-sorted algebras have been presented in the literature.

The definition of overloaded algebras that is used in this paper agrees mainly with [11, 13, 19, 20, 27, 31]. The algebras that Goguen et al. [16, 18] have described in some early papers are overloaded as well; however, in these papers homomorphisms are not families of functions but (as in the \mathcal{NCOV} -case) functions; note that this makes coherence unnecessary. An extension of the overloaded concept has been presented by Kreowski and Qian [29].

Our definition of nonoverloaded algebras is oriented chiefly towards [39]; however, we allow an operator symbol to have more than one arity. Similar descriptions can be found in [26, 35, 36, 37]. In Gogolla's papers [14, 15] and in Oberschelp's "einfache

mehrsortige Logik” (simple many-sorted logic) [33] operator symbols are interpreted by families of functions, but these can always be considered as restrictions of a single function; hence, these approaches must be regarded as nonoverloaded, too. Poigné [34] differs from the aforementioned authors in that he allows distinct sorts to have nondisjoint variable sets.

Oberschelp’s “mehrsortige Logik mit mehrfacher Interpretation der Prädikate und Funktionszeichen” (many-sorted logic with multiple interpretation of predicates and function symbols) [33] stands between the overloaded and nonoverloaded worlds.

The name “regularity” is used in different ways in papers on order-sorted algebras. In papers dealing with $\mathcal{L}\mathcal{C}\mathcal{V}$ -algebras a signature (S, \leq, Σ) is in general called regular if and only if every term $t \in T_{\Sigma}(X)$ has a least sort. This property proves to be very useful for computing unifiers in both the overloaded and the nonoverloaded case, as demonstrated in Subsection 4.1, but it is neither necessary nor sufficient for the overloaded term algebra to be initial: regularity and $\mathcal{L}\mathcal{V}$ -consistency do not imply each other.

Example 2.39. The signature Σ_1 is strongly consistent, but not regular.

$$(S_1, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma_1 = \{a: \rightarrow s_0, a: \rightarrow s_1, a: \rightarrow s_2\}.$$

The signature Σ_2 is regular, but not consistent.

$$(S_2, \leq) = \{s_3 \leq s_1 \leq s_0, s_3 \leq s_2 \leq s_0\},$$

$$\Sigma_2 = \{a: \rightarrow s_3, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0\}.$$

In the overloaded world regularity often means a substantially stronger property, which we will call GJM-regularity to avoid misunderstanding. A signature (S, \leq, Σ) is called GJM-regular [16] if for every $w^\circ \in S^*$, so that $f: w \rightarrow s$ and $w^\circ \leq w$, there exists a least $w's'$ satisfying $f: w' \rightarrow s'$ and $w^\circ \leq w'$. The GJM-regularity of a signature Σ implies the regularity of Σ ; its real importance is, however, that it implies the strong $\mathcal{L}\mathcal{V}$ -consistency of Σ and, thus, the initiality of the overloaded term algebra. Indeed, regularity and strong consistency together are still weaker than GJM-regularity, such that, even if regularity is actually needed, GJM-regularity generally poses an unnecessarily strong restriction. Note for instance that in this paper GJM-regularity is not needed at all and that the same is right for Ganzinger’s translation of order-sorted specifications to conditional many-sorted specifications [11].

Example 2.40. The signature Σ_3 is both strongly consistent and regular, but not GJM-regular:

$$(S_3, \leq) = \{s_3 \leq s_1 \leq s_0, s_3 \leq s_2 \leq s_0\},$$

$$\Sigma_3 = \{a: \rightarrow s_3, f: s_0 \rightarrow s_0, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0\}.$$

Unlike GJM-regularity, consistency is semantically defined. This has the further advantage that it can be easily adapted to other types of order-sorted algebras (e.g. [10, 33]), always yielding a sufficient criterion for the initiality of the term algebra.

3. Rewriting

3.1. Rewrite relations

An equation $(\forall Y)l \approx r$ is called a rewrite rule if we want to indicate that it should be used operationally in a specific direction. In order to express this notationally, we shall in general write $(\forall Y)l \rightarrow r$ instead of $(\forall Y)l \approx r$; if $Y = \text{Var}(l) \cup \text{Var}(r)$, we may omit the variable set Y . (We do not restrict to rules where $\text{Var}(r) \subseteq \text{Var}(l)$, following [6]).

Nevertheless, rewrite rules can be used as undirected equations, e.g. if R is a set of rewrite rules, we can ignore their orientation and write $t =_R^X t'$; on the other hand, we may sometimes regard “ordinary” equations as rewrite rules.

Definition 3.1. Let R be a set of rewrite rules. A term $t \in T_{\Sigma}(X)$ rewrites to $t' \in T_{\Sigma}(X)$ with a rewrite rule $(\forall Y)l \rightarrow r$ in R at the position $p \in \text{Pos}(t)$ if the following conditions are satisfied:

- (i) There exists a substitution $\sigma: Y \rightarrow T_{\Sigma}(X)$ such that $\sigma l = t/p$.
- (ii) $t' = t[p \leftarrow \sigma r]$.
- (iii) (a) In the overloaded case: there exists a sort $s \in S$ such that $t[p \leftarrow (x:s)]$ is a well-formed term and $\sigma l, \sigma r \in T_{\Sigma}(X)_s$.
- (b) In the nonoverloaded case: t' is a well-formed term.

We abbreviate this by $t \xrightarrow[\ell \gamma]{X} r$ or $t \xrightarrow[\ell \gamma]{X} r$, respectively. Sometimes we are also interested in the rule or in the position; in this case we write $t \xrightarrow[\ell \gamma]{X, [p, l \rightarrow r]} t'$ or $t \xrightarrow[\ell \gamma]{X, [p, l \rightarrow r]} t'$. (The application of an equation $(\forall Y)l \approx r$ to a term is defined analogously.)

Certain axioms, such as commutativity, are in general not turned into rewrite rules since they would cause the rewrite relation not to terminate. Instead, the set of terms is partitioned into congruence classes modulo these equations, and one uses a rewrite relation on the congruence classes.

Definition 3.2. The relation $\xrightarrow[\ell \gamma]{X}_{R/E}$ is defined by $\xrightarrow[\ell \gamma]{X}_{R/E} := \xrightarrow[\ell \gamma]{X}_R \circ \xrightarrow[\ell \gamma]{X}_R \circ \xrightarrow[\ell \gamma]{X}_E$. Analogously, $\xrightarrow[\ell \gamma]{X}_{R/E} := \xrightarrow[\ell \gamma]{X}_E \circ \xrightarrow[\ell \gamma]{X}_R \circ \xrightarrow[\ell \gamma]{X}_R$.

Rewriting using the relation $\rightarrow_{R/E}^X$ has some serious disadvantages. It may be very inefficient to determine whether a term t can be reduced by $\rightarrow_{R/E}^X$; the question may even be undecidable if the E -equivalence class of t is infinite. Thus, we now consider the weaker relation \rightarrow_E^X .

Definition 3.3. A term $t \in \mathsf{T}_\Sigma(X)$ rewrites to $t' \in \mathsf{T}_\Sigma(X)$ with a rewrite rule $(\forall Y)l \rightarrow r$ in R at the position p modulo E if the following conditions are satisfied.

- (i) There exists a substitution $\sigma: Y \rightarrow \mathsf{T}_\Sigma(X)$ such that $\sigma l \stackrel{X}{=}_{E} t/p$ or $\sigma l \stackrel{X}{=}_{E} t'/p$, respectively.
- (ii) $t' = t[p \leftarrow \sigma r]$.
- (iii) (a) In the overloaded case: there exist sorts $s, s' \in \mathcal{S}$ such that $t[p \leftarrow (x:s)]$ and $t[p \leftarrow (x:s')]$ are well-formed terms and $t/p, \sigma l \in \mathsf{T}_\Sigma(X)_s$ and $\sigma l, \sigma r \in \mathsf{T}_\Sigma(X)_{s'}$.
 (b) In the nonoverloaded case: $t[p \leftarrow \sigma l]$ and t' are well-formed terms.

We abbreviate this by $t \xrightarrow[\ell \gamma]{X} E \cdot R t'$ or $t \xrightarrow[\ell \gamma]{X} E \cdot R t$, respectively.

We can easily prove by induction on the length of p that $t \xrightarrow[\ell \gamma]{X} R^{[p, l \rightarrow r]} t'$ implies $t \stackrel{X}{=}_{R} t'$; thus, we know that $\xrightarrow[\ell \gamma]{X} R \subseteq \xrightarrow[\ell \gamma]{X} E \cdot R \subseteq \xrightarrow[\ell \gamma]{X} R \cdot E \subseteq \xrightarrow[\ell \gamma]{X} R \cup E$ (analogously in the nonoverloaded case). However, we encounter a crucial difference between unsorted (or many-sorted) and order-sorted rewriting. As demonstrated by Smolka et al. [39], given an order-sorted rewrite system R the relations $\stackrel{X}{=} R$ and $\overset{*}{\leftrightarrow} R$ are no longer guaranteed to be equal.

Example 3.4.

$$(\mathcal{S}, \leq) = \{s_1 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_1, b: \rightarrow s_1, c: \rightarrow s_0, f: s_1 \rightarrow s_1\},$$

$$R = \{a \rightarrow c, b \rightarrow c\}.$$

In the overloaded as well as in the nonoverloaded case $f(a) \stackrel{0}{=} R f(b)$ holds, but not $f(a) \overset{*}{\leftrightarrow} R f(b)$. As $f(c)$ is not a well-formed term, there cannot exist a $\overset{*}{\leftrightarrow} R$ -derivation of this equation. Note that R is even confluent.

3.2. Sort-decreasingness

Definition 3.5. A rewrite rule $(\forall Y)l \rightarrow r$ is called sort-decreasing if for all sorts $s \in \mathcal{S}$ and all substitutions $\sigma: Y \rightarrow \mathsf{T}_\Sigma(X)$ we have

$$\sigma l \in \mathsf{T}_\Sigma(X)_s \Rightarrow \sigma r \in \mathsf{T}_\Sigma(X)_s.$$

An equation $(\forall Y)l \approx r$ is called sort-preserving if for all sorts $s \in S$ and all substitutions $\sigma: Y \rightarrow T_{\Sigma}(X)$ we have

$$\sigma l \in T_{\Sigma}(X)_s \Leftrightarrow \sigma r \in T_{\Sigma}(X)_s.$$

A sort-decreasing rule may be applied to a term t whenever its left-hand side matches with a subterm of t since both the conditions (iii)(a) and (iii)(b) in Definition 3.1 are trivially satisfied. Besides if $t \in T_{\Sigma}(X)_s$ rewrites to t' with a sort-decreasing rule, then t' is in $T_{\Sigma}(X)_s$ as well.

Lemma 3.6. *Let X be a variable set containing at least one variable of every sort. Let Q be a subset of $T_{\Sigma}(X)$ such that for every spectrum $M \in \text{spctr}_{\Sigma \cup X}$ there is a term $t \in Q$ satisfying $M = \text{spctr}(t)$. Then the following two properties are equivalent:*

- (i) *The rewrite rule $(\forall Y)l \rightarrow r$ is sort-decreasing.*
- (ii) *For all substitutions $\sigma: \text{Var}(l) \cup \text{Var}(r) \rightarrow Q$ we have $\text{spctr}(\sigma l) \subseteq \text{spctr}(\sigma r)$.*

Proof. Let $V = \text{Var}(l) \cup \text{Var}(r)$. We first prove that (i) implies (ii). Suppose that $\sigma: V \rightarrow Q$ is a substitution and that the sort s is contained in $\text{spctr}(\sigma l)$. This implies $\sigma l \in T_{\Sigma}(X)_s$ and, as the rewrite rule $(\forall Y)l \rightarrow r$ is sort-decreasing, we get $s \in \text{spctr}(\sigma r)$.

To prove the reverse direction assume that $\sigma' l \in T_{\Sigma}(X)_s$ for an arbitrary substitution $\sigma': Y \rightarrow T_{\Sigma}(X)$ and a sort $s \in S$. Now choose a substitution $\sigma: V \rightarrow Q$ such that $\text{spctr}(\sigma x) = \text{spctr}(\sigma' x)$ for every variable $x \in V$. By induction on the term structure it can be proved that $\text{spctr}(\sigma t) = \text{spctr}(\sigma' t)$ for all terms $t \in T_{\Sigma}(V)$; so, we have $s \in \text{spctr}(\sigma' l)$ if and only if $s \in \text{spctr}(\sigma l)$. By property (ii) this implies $s \in \text{spctr}(\sigma r)$; thus, $s \in \text{spctr}(\sigma' r)$ and $\sigma' r \in T_{\Sigma}(X)_s$. \square

Theorem 3.7. *For finite signatures the sort-decreasingness of a rewrite rule is decidable.*

Proof. If the signature (S, \leq, Σ) is finite, we can compute a finite set $Q \subseteq T_{\Sigma}(X)$ having the property described above according to Lemma 2.5. As there are only finitely many substitutions $\sigma: \text{Var}(l) \cup \text{Var}(r) \rightarrow Q$, sort-decreasingness is decidable. \square

Corollary 3.8. *For finite signatures the sort-preservingness of an equation is decidable.*

Proof. As in Lemma 3.6 we show that $(\forall Y)l \approx r$ is sort-preserving if and only if $\text{spctr}(\sigma l) = \text{spctr}(\sigma r)$ holds for every substitution $\sigma: \text{Var}(l) \cup \text{Var}(r) \rightarrow Q$. (The set Q is chosen as in Theorem 3.7.) \square

Lemma 3.9. *Let R be a set of sort-decreasing rules. Then the overloaded and the nonoverloaded rewrite relations coincide, i.e. we have $t \xrightarrow[\text{overloaded}]{X} t'$ if and only if $t \xrightarrow[\text{nonoverloaded}]{X} t'$.*

Proof. If every rule in R is sort-decreasing, both the conditions (iii) (a) and (iii)(b) in Definition 3.1 are trivially satisfied. \square

Lemma 3.10. *Let E be a set of sort-preserving equations. Then E induces in the overloaded and in the nonoverloaded case the same notion of equality on $T_\Sigma(X)$; we have $t \stackrel{X}{\underset{\mathcal{C}\mathcal{Y}}{E}} t'$ if and only if $t \stackrel{X}{\underset{\mathcal{A}\mathcal{C}\mathcal{Y}}{E}} t'$.*

Proof. First we show that in the overloaded as well as in the nonoverloaded case $E \vdash t \approx t'$ implies $\text{spctr}(t) = \text{spctr}(t')$. This is easily proved by induction on the length of the derivation. For the rules (E1)–(E3) it is trivial; for rule (E4a) or (E4b) it follows from the recursion formula to compute $\text{spctr}(f(t_1, \dots, t_n))$; for rule (E5) it results from the sort-preservingness of the equations in E .

Now we consider the two rules (E4a) and (E4b) once more. Since all equations $(\forall X)t_i \approx t'_i$ have the property that $\text{spctr}(t_i) = \text{spctr}(t'_i)$, the additional conditions of rule (E4a) and (E4b) are equivalent. Both inference rules coincide; hence, $\stackrel{X}{\underset{\mathcal{C}\mathcal{Y}}{E}}$ equals $\stackrel{X}{\underset{\mathcal{A}\mathcal{C}\mathcal{Y}}{E}}$. \square

Corollary 3.11. *Let R be a set of sort-decreasing rules and E be a set of sort-preserving equations. Then $\xrightarrow{\mathcal{C}\mathcal{Y}}_{R/E}^X$ equals $\xrightarrow{\mathcal{A}\mathcal{C}\mathcal{Y}}_{R/E}^X$ and $\xrightarrow{\mathcal{C}\mathcal{Y}}_{E \setminus R}^X$ equals $\xrightarrow{\mathcal{A}\mathcal{C}\mathcal{Y}}_{E \setminus R}^X$.*

Provided that the conditions of the above lemmas are fulfilled, we can omit the indices $\mathcal{C}\mathcal{Y}$ and $\mathcal{A}\mathcal{C}\mathcal{Y}$ at the relations \xrightarrow{X}_R , $\xrightarrow{X}_{E \setminus R}$, $\xrightarrow{X}_{R/E}$ and $=^X_E$. However, this is not correct for the relations $=^X_R$ and $=^X_{R \cup E}$, as the following example demonstrates.

Example 3.12. Let Σ and R be defined by

$$(S, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_0, b: \rightarrow s_1, c: \rightarrow s_2, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0\},$$

$$R = \{a \rightarrow b, a \rightarrow c\}.$$

We have $f(b) \stackrel{\emptyset}{\underset{\mathcal{A}\mathcal{C}\mathcal{Y}}{R}} f(c)$, but not $f(b) \stackrel{\emptyset}{\underset{\mathcal{C}\mathcal{Y}}{R}} f(c)$. Note that all rules in R are sort-decreasing.

Finally it should be mentioned explicitly that even now all $\mathcal{C}\mathcal{Y}$ -relations are only defined for coherent and strongly consistent signatures.

3.3. Confluence and Church–Rosser property

From now on we consider only sort-decreasing rewrite rules; moreover, we assume that all equations from E are sort-preserving.

Definition 3.13. A rewrite system (R, E) is said to be Church–Rosser modulo E , if the relations $=^X_{R \cup E}$ and $\overset{*}{\xrightarrow{X}_{R/E}} \circ =^X_E \circ \overset{*}{\xleftarrow{X}_{R/E}}$ agree.

Definition 3.14. A rewrite system (R, E) is said to be confluent modulo E if the relation $\xrightarrow{*}_{R/E}^X \circ \xrightarrow{*}_{R/E}^X$ is included in $\xrightarrow{*}_{R/E}^X \circ \xrightarrow{*}_E^X \circ \xrightarrow{*}_{R/E}^X$.

Theorem 3.15. Let R be a set of sort-decreasing rules and E a set of sort-preserving equations. The rewrite system (R, E) has the Church–Rosser property modulo E if and only if it is confluent modulo E .

Proof. Since $t \xrightarrow{*}_{R/E}^X t'$ implies $t =_{R \cup E}^X t'$, it suffices to show that the Church–Rosser property follows from confluence; this can be proved by induction on the length of the derivation of $t =_{R \cup E}^X t'$ according to Inference system 2.30. \square

Corollary 3.16. For a set E of sort-preserving equations the two relations $=_E^X$ and $\xrightarrow{*}_E^X$ agree.

Proof. Let E^{-1} be defined by $E^{-1} := \{(\forall Y)r \approx l \mid (\forall Y)l \approx r \in E\}$. Now let R denote $E \cup E^{-1}$, then (R, \emptyset) is a confluent rewrite system and $t_1 =_E^X t_2$ implies $t_1 =_{R \cup \emptyset}^X t_2$. By Theorem 3.15 there are terms $t_3, t_4 \in \mathbb{T}_\Sigma(X)$ such that $t_1 \xrightarrow{*}_{R, \emptyset}^X t_3 =_{\emptyset}^X t_4 \xrightarrow{*}_{R, \emptyset}^X t_2$; this implies $t_1 \xrightarrow{*}_R^X t_3 = t_4 \xrightarrow{*}_R^X t_2$ and, thus, $t_1 \xrightarrow{*}_E^X t_2$. The proof of the reverse direction is trivial. \square

Corollary 3.17. Let R be a set of sort-decreasing rules and let E be a set of sort-preserving equations such that (R, E) is confluent modulo E . Then the following properties are equivalent:

- (i) $t_1 \xrightarrow[\ell \gamma]{*}_{R \cup E}^X t_2$.
- (ii) $t_1 \xrightarrow[\ell \gamma]{*}_{R \cup E}^X t_2$.
- (iii) There are $t'_1, t'_2 \in \mathbb{T}_\Sigma(X)$ such that $t_1 \xrightarrow[\ell \gamma]{*}_{R/E}^X t'_1 \xrightarrow[\ell \gamma]{*}_E^X t'_2 \xrightarrow[\ell \gamma]{*}_{R/E}^X t_2$.
- (iv) $t_1 \xrightarrow[\ell \gamma]{*}_{R \cup E}^X t_2$.
- (v) $t_1 \xrightarrow[\ell \gamma]{*}_{R \cup E}^X t_2$.
- (vi) There are $t'_1, t'_2 \in \mathbb{T}_\Sigma(X)$ such that $t_1 \xrightarrow[\ell \gamma]{*}_{R/E}^X t'_1 \xrightarrow[\ell \gamma]{*}_E^X t'_2 \xrightarrow[\ell \gamma]{*}_{R/E}^X t_2$.

Proof. Obviously, (ii) implies (i) and (v) implies (iv). Since $=_E^X = \xrightarrow{*}_E^X$, we know that $\xrightarrow{*}_{R/E}^X \subseteq \xrightarrow{*}_{R \cup E}^X$; so, (iii) implies (ii) and (vi) implies (v). By Theorem 3.15 the properties (i) and (iii) and the properties (iv) and (vi) are equivalent; finally, the equivalence of (iii) and (vi) follows from $\xrightarrow[\ell \gamma]{*}_R^X = \xrightarrow[\ell \gamma]{*}_{R \cup E}^X$ and $\xrightarrow[\ell \gamma]{*}_E^X = \xrightarrow[\ell \gamma]{*}_{R \cup E}^X$. \square

In the following definitions \mathcal{R} replaces R or $E \setminus R$ or R/E .

Definition 3.18. We say that the rewrite relation $\rightarrow_{\mathcal{R}}^X$ terminates if there is no infinite sequence (t_1, t_2, \dots) where $t_i \rightarrow_{\mathcal{R}}^X t_{i+1}$.

Definition 3.19. A term t is called irreducible under $\rightarrow_{\mathcal{R}}^X$ if there is no term t' satisfying $t \rightarrow_{\mathcal{R}}^X t'$; we also say that t is in \mathcal{R} -normal form. A term t has the \mathcal{R} -normal form t' if $t \xrightarrow{*}_{\mathcal{R}}^X t'$ holds and t' is irreducible under \mathcal{R} .

If the rewrite relation $\rightarrow_{\mathcal{R}}^X$ terminates, every term $t \in T_{\Sigma}(X)$ has at least one \mathcal{R} -normal form.

Definition 3.20. A rewrite system (R, E) is called $(E \setminus R)$ -Church–Rosser modulo E if the relations $=_{R \cup E}^X$ and $\xrightarrow{*}_{E \setminus R}^X \circ \xrightarrow{*}_{E}^X \circ \xrightarrow{*}_{E \setminus R}^X$ agree.

As in the unsorted case, we can define the global and local confluence and coherence of the relation $\rightarrow_{E \setminus R}^X$ (see [25] for details). The relations between these properties are expressed in the following theorem [13].

Theorem 3.21. Let R be a set of sort-decreasing rules and let E be a set of sort-preserving equations. Provided that the relation $\rightarrow_{R/E}^X$ terminates, the following properties are equivalent:

- (i) (R, E) is $(E \setminus R)$ -Church–Rosser modulo E .
- (ii) The relations $\xrightarrow{*}_{R \cup E}^X$ and $\xrightarrow{*}_{E \setminus R}^X \circ \xrightarrow{*}_{E}^X \circ \xrightarrow{*}_{E \setminus R}^X$ agree.
- (iii) The relation $\rightarrow_{E \setminus R}^X$ is confluent modulo E and coherent modulo E .
- (iv) The relation $\rightarrow_{E \setminus R}^X$ is locally confluent modulo E with respect to R and locally coherent modulo E .
- (v) For all terms $t, t' \in T_{\Sigma}(X)$ we have $t \xrightarrow{*}_{R \cup E}^X t'$ if and only if there are normal forms $t_{\text{NF}}, t'_{\text{NF}}$ of t and t' with respect to $\rightarrow_{E \setminus R}^X$ such that $t_{\text{NF}} \xrightarrow{*}_{E}^X t'_{\text{NF}}$.

Proof. As in Jouannaud’s and Kirchner’s unsorted version of this theorem [25], the proof consists of the implications (i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (iv) \Rightarrow (v) \Rightarrow (i). Two minor differences from the unsorted version are due to the fact that in order-sorted rewriting the relations $=_{R \cup E}^X$ and $\xrightarrow{*}_{R \cup E}^X$ do not necessarily agree. In the first place, property (i) and property (ii) do no longer collapse. Secondly, the proof of the (v) \Rightarrow (i) part requires now a simple induction on the length of the derivation of $t =_{R \cup E}^X t'$ according to Inference system 2.30. \square

3.4. Remarks

Condition (iii)(a) of Definition 3.3 seems to be rather complicated. Several authors have presented apparently easier approaches.

For instance one might replace (iii)(a) by one of the following three properties:

- (iii) (a') There exists a sort $s \in \mathcal{S}$ such that $t[p \leftarrow (x : s)]$ is a well-formed term and $\sigma l, \sigma r \in T_{\Sigma}(X)_s$.

- (iii) (a)'' There exists a sort $s \in S$ such that $t[p \leftarrow (x : s)]$ is a well-formed term and $t/p, \sigma r \in T_{\Sigma}(X)_s$.
- (iii) (a)''' There exists a sort $s \in S$ such that $t[p \leftarrow (x : s)]$ is a well-formed term and $t/p, \sigma l, \sigma r \in T_{\Sigma}(X)_s$.

Replacing (iii)(a) by (iii)(a)', however, renders the relation $\xrightarrow[\ell \gamma]{X} E \setminus R$ unsound.

Example 3.22.

$$(S, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma = \{a : \rightarrow s_1, b : \rightarrow s_2, c : \rightarrow s_2, f : s_1 \rightarrow s_0, f : s_2 \rightarrow s_0\},$$

$$E = \{a \approx b\},$$

$$R = \{b \rightarrow c\}.$$

We have $f(a) \xrightarrow[\ell \gamma]{\emptyset} E \setminus R} f(c)$, but $f(a) \not\xrightarrow[\ell \gamma]{\emptyset} R \cup E} f(c)$ does not hold.

If (iii)(a) is replaced by (iii)(a)'', the relation $\xrightarrow[\ell \gamma]{X} E \setminus R$ is sound, but $\xrightarrow[\ell \gamma]{X} E \setminus R$ is no longer included in $\xrightarrow[\ell \gamma]{X} R/E$.

Example 3.23.

$$(S, \leq) = \{s_1 \leq s_0\},$$

$$\Sigma = \{a : \rightarrow s_1, b : \rightarrow s_1, c : \rightarrow s_0, f : s_1 \rightarrow s_0\},$$

$$E = \{a \approx c\},$$

$$R = \{c \rightarrow b\}.$$

We have $f(a) \xrightarrow[\ell \gamma]{\emptyset} E \setminus R} f(b)$, but $f(a) \not\xrightarrow[\ell \gamma]{\emptyset} R/E} f(b)$ does not hold.

It is less problematic to replace (iii)(a) by (iii)(a)'''. This condition is even stronger than (iii)(a); so, we have $\xrightarrow[\ell \gamma]{X} E \setminus R} \subseteq \xrightarrow[\ell \gamma]{X} R/E} \subseteq \xrightarrow[\ell \gamma]{X} R \cup E}$. A drawback of this approach, however, is that the definition of compatibility becomes more complicated.

The decidability of the sort-decreasingness property was already stated in [39, 36]. The criterion mentioned in [39], however, is only correct for regular signatures. In general, it is not sufficient to check all specializations in order to decide whether a rewrite rule is sort-decreasing or not.

Example 3.24.

$$(S, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma = \{a : \rightarrow s_0, a : \rightarrow s_1, a : \rightarrow s_2, b : \rightarrow s_0, f : s_0 s_0 \rightarrow s_0, f : s_1 s_2 \rightarrow s_1\},$$

$$R = \{(\forall \{x : s_0\}) f(x, x) \rightarrow b\}.$$

As $\text{spctr}(f(a, a)) = \{s_0, s_1\} \not\subseteq \{s_0\} = \text{spctr}(b)$, the rewrite rule in R is not sort-decreasing. On the other hand, we have $\text{spctr}(f(x_0:s_0, x_0:s_0)) = \text{spctr}(f(x_1:s_1, x_1:s_1)) = \text{spctr}(f(x_2:s_2, x_2:s_2)) = \{s_0\} = \text{spctr}(b)$; hence, $\text{spctr}(\sigma f(x:s_0, x:s_0)) \subseteq \text{spctr}(\sigma b)$ holds for every specialization σ , and R satisfies the criterion from [39].

The criterion that was given in [36] depends on the assumption that for every $s \in S$ and $t \in T_{\Sigma}(X)$ the set of substitutions $\sigma: X \rightarrow T_{\Sigma}(Y)$ satisfying $\sigma t \in T_{\Sigma}(Y)_s$ has a finite subset that is complete with respect to the subsumption ordering. The following example contradicts this assumption.

Example 3.25.

$$\begin{aligned} (S, \leq) &= \{s_2 \leq s_1, s'_2 \leq s_1, s_1 \leq s_0\}, \\ \Sigma &= \{a: \rightarrow s_0, a: \rightarrow s_2, a: \rightarrow s'_2, g: s_1 s_1 \rightarrow s_0, \\ & f: s_0 \rightarrow s_0, f: s_2 \rightarrow s_2, f: s'_2 \rightarrow s'_2, g: s_2 s'_2 \rightarrow s_1\}. \end{aligned}$$

Let $t := g(x:s_1, x:s_1)$ and $s := s_1$. For $n \geq 0$ we define the substitution σ_n by $\sigma_n := \{x:s_1 \mapsto f^n(a)\}$; then $\sigma_n t \in T_{\Sigma}(Y)_{s_1}$ holds for every $n \geq 0$. The set of all σ_n is complete and minimal, yet infinite. So, there cannot exist a finite and complete set of substitutions θ such that $\theta t \in T_{\Sigma}(Y)_{s_1}$.

Theorems 3.15 and 3.21 remain valid if we dispense with sort-decreasingness and sort-preservingness and only require that rules and equations be compatible. Intuitively, compatibility means that whenever a rewrite rule may be applied to a term t , this application is also possible if t is a subterm of some larger term.

Definition 3.26. A rewrite rule $(\forall Y)l \rightarrow r$ is called \mathcal{CV} -compatible if for every term $t \in T_{\Sigma}(X)$ we have: If there is a substitution $\sigma: Y \rightarrow T_{\Sigma}(X)$ and a position $p \in \text{Pos}(t)$ and if $\sigma l = t/p$, then there exists a sort $s \in S$ such that $t[p \leftarrow (x:s)]$ is a well-formed term and both σl and σr are in $T_{\Sigma}(X)_s$.

A set E of equations is called \mathcal{CV} -compatible if for every term $t \in T_{\Sigma}(X)$ we have: If there is a term t' and a position $p \in \text{Pos}(t)$ and if $t/p \stackrel{X}{\underset{E}{\equiv}} t'$, then there exists a sort $s \in S$ such that $t[p \leftarrow (x:s)]$ is a well-formed term and both t/p and t' are in $T_{\Sigma}(X)_s$.

Definition 3.27. A rewrite rule $(\forall Y)l \rightarrow r$ is called \mathcal{ACV} -compatible if for every term $t \in T_{\Sigma}(X)$ we have: If there is a substitution $\sigma: Y \rightarrow T_{\Sigma}(X)$ and a position $p \in \text{Pos}(t)$ and if $\sigma l = t/p$, then $t[p \leftarrow \sigma r]$ is a well-formed term.

An equation $(\forall Y)l \approx r$ is called \mathcal{ACV} -compatible if both the rules $(\forall Y)l \rightarrow r$ and $(\forall Y)r \rightarrow l$ are \mathcal{ACV} -compatible.

As proved in [42] the $\mathcal{N}\mathcal{C}\mathcal{V}$ -compatibility of a rewrite rule is decidable; besides Smolka et al. [39] have demonstrated that it is easy to make a signature $\mathcal{N}\mathcal{C}\mathcal{V}$ -compatible by construction (add declarations $f: \top \dots \top \rightarrow \top$ for a new, greatest sort \top).

The following example shows that in the overloaded case the weaker notion of $\mathcal{N}\mathcal{C}\mathcal{V}$ -compatibility is not sufficient for Theorem 3.15 to hold.

Example 3.28.

$$(S, \leq) = \{s_1 \leq s_0, s_2 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_1, b: \rightarrow s_1, c: \rightarrow s_2, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0\},$$

$$R = \{a \rightarrow c, b \rightarrow c\}.$$

The set R is $\mathcal{N}\mathcal{C}\mathcal{V}$ -compatible and confluent. Now we have $f(a) \xrightarrow[\mathcal{C}\mathcal{V}]{\emptyset} f(b)$, but not $f(a) \xrightarrow[\mathcal{C}\mathcal{V}]{*} f(b)$. As $f(a)$ and $f(c)$ are not R -equal, there cannot exist an overloaded $\xrightarrow[\mathcal{C}\mathcal{V}]{*}$ -derivation for this equation.

4. Completion

4.1. Unification and regularity

From now on we always assume that (S, \leq, Σ) is a signature such that there is a ground term $t \in T_{\Sigma, s}$ for every sort $s \in S$.

Definition 4.1. Let $t, t' \in T_{\Sigma}(X)$. We say that a substitution $\sigma: X \rightarrow T_{\Sigma}(Y)$ is a unifier of t and t' if $\sigma t = \sigma t'$ holds. A substitution $\sigma: X \rightarrow T_{\Sigma}(Y)$ is called an E -unifier of t and t' if $\sigma t =_E \sigma t'$ holds. The set of all unifiers of t and t' is denoted by $\text{SU}(t, t')$; $\text{SU}_E(t, t')$ represents the set of all E -unifiers of t and t' .

Definition 4.2. Let $t, t' \in T_{\Sigma}(X)$. A subset $U \subseteq \text{SU}(t, t')$ is called complete if for every unifier $\sigma': X \rightarrow T_{\Sigma}(Y')$ there exists a $\sigma: X \rightarrow T_{\Sigma}(Y)$ from U and a substitution $\theta: Y \rightarrow T_{\Sigma}(Y')$ such that $\sigma' = \theta \circ \sigma$. Analogously, a subset $U \subseteq \text{SU}_E(t, t')$ is called complete if for every E -unifier $\sigma': X \rightarrow T_{\Sigma}(Y')$ there exists a $\sigma: X \rightarrow T_{\Sigma}(Y)$ from U and a substitution $\theta: Y \rightarrow T_{\Sigma}(Y')$ such that $\sigma' =_E \theta \circ \sigma$, i.e. $\sigma' x =_E \theta(\sigma x)$ for each $x \in X$.

Definition 4.3. A complete set U of unifiers or E -unifiers of t and t' is said to be minimal if it does not contain two distinct substitutions σ, σ' so that $\sigma' = \theta \circ \sigma$ or $\sigma' =_E \theta \circ \sigma$, respectively, for some substitution θ .

A complete set of unifiers is called a CSU, a minimal and complete set of unifiers is called a μ CSU.

Overloaded and nonoverloaded semantics do not differ with respect to unification if we consider syntactical unification or unification with respect to a set of sort-preserving equations. On the other hand, if E contains an equation that is not sort-preserving, it may e.g. happen that overloaded E -unification is unitary and nonoverloaded E -unification is infinitary.

Unification in order-sorted signatures differs substantially from many-sorted unification. Even if the signature is finite and if we consider only syntactical unification (i.e. unification with respect to the empty set of equations), a minimal complete set of unifiers may be infinite, as demonstrated by the following example.

Example 4.4.

$$(\mathcal{S}, \leq) = \{s_1 \leq s_0, s'_1 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_0, a: \rightarrow s_1, a: \rightarrow s'_1, f: s_0 \rightarrow s_0, f: s_1 \rightarrow s_1, f: s'_1 \rightarrow s'_1\}.$$

Consider the terms $x: s_1$ and $x': s'_1$. For each $n \in \mathbb{N}$ we define the substitution σ_n by $\sigma_n := \{x: s_1 \mapsto f^n(a), x': s'_1 \mapsto f^n(a)\}$. Every σ_n is a unifier of x and x' and the set of all σ_n is complete and minimal, yet infinite. Since for any two minimal complete sets of unifiers U_1 and U_2 there is a bijection from U_1 to U_2 , the two terms x and x' cannot have a finite μ CSU.

In order to avoid the problems arising from infinite μ CSUs, it is useful to restrict the class of signatures to be considered.

Definition 4.5. A signature Σ is called regular if every term $t \in T_{\Sigma}(V)$ has a least sort, which is denoted by $\text{LS}(t)$, i.e. if every $M \in \text{spctr}_{\Sigma \cup V}$ has a smallest element. (We assume that V contains at least one variable of every sort.)

Since $\text{spctr}_{\Sigma \cup V}$ is finite and computable, the regularity of a finite signature is decidable (cf. [37]). In regular signatures for every term $t \in T_{\Sigma}(Y)$ there is a variable $x \in Y'$ such that $\text{spctr}(x) = \text{spctr}(t)$; similarly, for every substitution $\sigma: X \rightarrow T_{\Sigma}(Y)$ there is a specialization $\theta: X \rightarrow Y'$ such that $\text{spctr}(\sigma t) = \text{spctr}(\theta t)$ for every $t \in T_{\Sigma}(X)$. The subsequent theorem originating from Schmidt-Schauß [35] is an important consequence of this fact.

Theorem 4.6. *If $(\mathcal{S}, \leq, \Sigma)$ is a finite regular signature, then for any two terms from $T_{\Sigma}(X)$ a finite CSU is effectively computable.*

4.2. Critical pairs

We introduce the following conventions. In the rest of this section (S, \leq, Σ) is always a finite and regular signature such that for every sort $s \in S$ there is at least one ground term $t \in T_{\Sigma, s}$. (By Lemma 2.28 this implies that the variable set may be omitted in the notation of applications of rules and equations.) We consider only sort-decreasing rewrite rules $l \rightarrow r$ such that $\text{Var}(r) \subseteq \text{Var}(l)$ and such that l is not a variable. The set E consists of sort-preserving Σ -equations $t \approx t'$ such that t and t' contain the same variables; neither t nor t' is a variable, and the strict subterm ordering modulo E is noetherian. In order to simplify notation we may assume without loss of generality that E is symmetrical. Unless explicitly stated otherwise, all considerations are valid in both the overloaded and the nonoverloaded case.

The notion of E -critical pairs was introduced by Jouannaud. The following lemma originates from [24].

Definition 4.7. Let $(\forall X)g \rightarrow d$ and $(\forall X)l \rightarrow r$ be two rewrite rules such that $\text{Var}(g)$ and $\text{Var}(l)$ are disjoint and let p be a position in g such that g/p is not a variable. If U is a complete set of E -unifiers of g/p and l , we call the set

$$\{(\sigma d, (\sigma g)[p \leftarrow \sigma r]) \mid \sigma \in U\}$$

a complete set of E -critical pairs of the rule $g \rightarrow d$ with the rule $l \rightarrow r$ at the position p . (Analogously for an equation $(\forall X)g \approx d$ and a rewrite rule $(\forall X)l \rightarrow r$.)

Lemma 4.8. Let $t, t_1, t_2 \in T_{\Sigma}(Y)$ and let $(\forall X)g \rightarrow d$ and $(\forall X)l \rightarrow r$ be two rules in R such that $\text{Var}(g) \cap \text{Var}(l) = \emptyset$ and $t_1 \leftarrow_R^{[o, g \rightarrow d]} t \rightarrow_{E \setminus R}^{[op, l \rightarrow r]} t_2$, where $p \in \text{Pos}(g)$ and $g/p \notin X$. Besides, let C be a complete set of E -critical pairs of the rule $g \rightarrow d$ with the rule $l \rightarrow r$ at the position p . Then there is a pair $(q_1, q_2) \in C$ and a substitution θ such that $t_1/o \xrightarrow{*}_E \theta q_1$ and $t_2/o \xrightarrow{*}_E \theta q_2$. (Analogously for an equation $(\forall X)g \approx d$ and a rewrite rule $(\forall X)l \rightarrow r$.)

4.3. A completion procedure

A sort-decreasing term rewrite system is called complete if it terminates and has the Church–Rosser property (or $(E \setminus R)$ -Church–Rosser property). Completion is the transformation of a set of equations into an equivalent complete rewrite system. The first completion algorithm was presented by Knuth and Bendix [28]. The idea that the classical completion method can also be used for order-sorted specifications, provided that all generated rules are sort-decreasing, is due to Gnaedig et al. [13].

Definition 4.9. A strict ordering $>$ on $T_{\Sigma}(X)$ is a reduction ordering if it fulfils the following conditions:

- (i) $>$ is a noetherian ordering.
- (ii) $t_1 > t_2$ implies $\sigma t_1 > \sigma t_2$ for all terms t_1, t_2 and all substitutions σ .
- (iii) Given terms t, t_1 and t_2 from $T_{\Sigma}(X)$ such that p is a position of t and that $t[p \leftarrow t_1]$ and $t[p \leftarrow t_2]$ are well-formed terms, $t_1 > t_2$ implies $t[p \leftarrow t_1] > t[p \leftarrow t_2]$.

Definition 4.10. We say that a reduction ordering \succ on $T_{\Sigma}(X)$ is compatible with E if $t'_1 =_E t_1 \succ t_2 =_E t'_2$ implies $t'_1 \succ t'_2$.

Reduction orderings have the following fundamental property. A rewrite relation \rightarrow_R is terminating if and only if R is contained in some reduction ordering \succ , i.e. if and only if $l \succ r$ holds for all rules $l \rightarrow r$ in R . Analogously, a rewrite relation $\rightarrow_{R/E}$ is terminating if and only if R is contained in a reduction ordering that is compatible with E [2]. In the rest of this section we always assume that \succ is a reduction ordering on $T_{\Sigma}(X)$ that is compatible with E .

The completion procedure is described as an inference system, where starting with a set of equations G_0 and a set of rules R_0 new pairs (G_i, R_i) are inferred such that the relation $=_{G_i \cup R_i \cup E}$ remains invariant. It is necessary that all rules $l \rightarrow r$ in R_0 are sort-decreasing and satisfy $l \succ r$; often, R_0 will be empty.

Inference system 4.11.

- (C1) Orienting an equation:
$$\frac{G \cup \{l \approx r\}, R}{G, R \cup \{l \rightarrow r\}}$$
if the rule $l \rightarrow r$ is sort-decreasing and $l \succ r$.
- (C2) Adding an equation:
$$\frac{G, R}{G \cup \{l \approx r\}, R}$$
if $l \xleftarrow{*}_{R \cup E} q \xrightarrow{*}_{R \cup E} r$.
- (C3) Simplifying an equation:
$$\frac{G \cup \{l \approx r\}, R}{G \cup \{q \approx r\}, R}$$
if $l \rightarrow_{R/E} q$.
- (C4) Deleting an equation:
$$\frac{G \cup \{l \approx r\}, R}{G, R}$$
if $l \xleftrightarrow{*}_E r$.
- (C5) Simplifying the right side:
$$\frac{G, R \cup \{l \rightarrow r\}}{G, R \cup \{l \rightarrow q\}}$$
if $r \rightarrow_{R/E} q$.
- (C6) Simplifying the left side:
$$\frac{G, R \cup \{l \rightarrow r\}}{G \cup \{q \approx r\}, R}$$
if $l \rightarrow_{R/E} q$ at a position $p \neq \varepsilon$ or if $l \rightarrow_R q$ using a rule $l' \rightarrow r'$ so that l and l' are not equal up to renaming of variables.

To simplify notation we write only one of the possible orientations of an equation $l \approx r$. However, all inference rules are also valid for the inverse equation $r \approx l$. Note that all the rewrite rules generated are sort-decreasing.

The limit of a derivation sequence $(G_0, R_0), (G_1, R_1), \dots$ is the pair (G_∞, R_∞) , where $G_\infty := \bigcup_i \bigcap_{j \geq i} G_j$ and $R_\infty := \bigcup_i \bigcap_{j \geq i} R_j$ denote the sets of persisting rules and persisting equations. By convention the index i ranges from 0 to n if the derivation is finite and has the length n ; otherwise, i ranges from 0 to ∞ .

The Inference system 4.11 is only a part of the completion algorithm that we want to describe. In fact, a sequence $(G_0, R_0), (G_1, R_1), \dots$ does not yield necessarily a complete term rewrite system; so, we need a fairness criterion to select suitable derivations (see [1] for details). As in the unsorted case, every fair derivation yields a complete term rewrite system. Most parts of the proof of this statement can be carried out as in Bachmair's unsorted version. The differences are explained in the sequel.

A proof P for an equation $t \approx t'$ over $G \cup R \cup E$ is a sequence (t_1, \dots, t_k) such that $t = t_1$ and $t' = t_k$ and for $1 \leq i \leq k$ either $t_{i-1} \leftrightarrow_G t_i$ or $t_{i-1} \rightarrow_R t_i$ or $t_{i-1} \leftarrow_R t_i$ or $t_{i-1} \leftrightarrow_E t_i$ holds.

As in [1], we define an ordering $>_c$ on proofs that is well-founded and monotonic with respect to the proof structure, with respect to the term structure and with respect to instantiation.

Each application of an inference rule is reflected by transformations of proofs. If (G', R') is obtained by an application of one of the inference rules (C1)–(C6) to (G, R) , then the generated congruences $=_{G \cup R \cup E}$ and $=_{G' \cup R' \cup E}$ are the same. Unlike in the unsorted case, however, the relations $\overset{*}{\leftrightarrow}_{G \cup R \cup E}$ and $\overset{*}{\leftrightarrow}_{G' \cup R' \cup E}$ are not necessarily equal; the same holds for $=_{G \cup R \cup E}$ and $\overset{*}{\leftrightarrow}_{G \cup R \cup E}$. We can merely show that $\overset{*}{\leftrightarrow}_{G \cup R \cup E}$ is a subset of $\overset{*}{\leftrightarrow}_{G' \cup R' \cup E}$.

Lemma 4.12. *Let $(G_0, R_0), (G_1, R_1), \dots$ be a derivation according to Inference system 4.11 and let P be a proof of $t_1 \approx t_k$ in $G_i \cup R_i \cup E$. If $j > i$ and if P is not a proof in $G_j \cup R_j \cup E$, then there exists a proof Q of $t_1 \approx t_k$ in $G_j \cup R_j \cup E$ satisfying $P >_c Q$.*

Noetherian induction on $>_c$ yields the following lemma.

Lemma 4.13. *Let $(G_0, R_0), (G_1, R_1), \dots$ be a derivation according to Inference system 4.11. Then for every proof P of $t_1 \approx t_k$ in $G_i \cup R_i \cup E$ there is a proof Q of $t_1 \approx t_k$ in $G_\infty \cup R_\infty \cup E$ such that $P \geq_c Q$.*

Lemma 4.14. *If G and G' are two sets of Σ -equations and $l \overset{G}{\equiv} r$ holds for every equation $l \approx r$ in G , then $\overset{G}{\equiv} \subseteq \overset{G'}{\equiv}$. Analogously, if $l \overset{G'}{\equiv} r$ holds for every equation $l \approx r$ in G , then $\overset{G}{\equiv} \subseteq \overset{G'}{\equiv}$.*

Proof. By induction on Inference system 2.30. \square

Using the two preceding lemmas we can prove now Lemma 4.15.

Lemma 4.15. *Let $(G_0, R_0), (G_1, R_1), \dots$ be a derivation according to Inference system 4.11. Then the relations $=_{G_i \cup R_i \cup E}$ and $=_{G_i \cup R_i \cup E}$ agree for every i .*

Theorem 4.16. *If $(G_0, R_0), (G_1, R_1), \dots$ is a fair derivation according to Inference system 4.11, then R_∞ is a complete term rewrite system.*

Proof. Bachmair's unsorted proof (by induction on the proof ordering $>_c$) [1] can be used as well for order-sorted completion. Note, however, that it does not yield the equality of $=_{R \cup E}$ and $\xrightarrow{*}_{E \setminus R} \circ \xleftarrow{*}_{E^\circ} \xleftarrow{*}_{E \setminus R}$, but only $\xleftrightarrow{*}_{R \cup E} = \xrightarrow{*}_{E \setminus R} \circ \xleftarrow{*}_{E^\circ} \xleftarrow{*}_{E \setminus R}$. We need Theorem 3.21 to see that these two properties are indeed equivalent. \square

4.4. Overloaded versus nonoverloaded completion

The Gnaedig–Kirchner–Kirchner completion procedure described above may be used in both the overloaded and the nonoverloaded case. Indeed, we can prove that every overloaded completion procedure has this property.

Theorem 4.17. *If G and G' are two sets of Σ -equations and $\xrightarrow[\ell \tau]{=} G = \xrightarrow[\ell \tau]{=} G'$ holds, then we have also $\xrightarrow[\ell \tau]{=} G = \xrightarrow[\ell \tau]{=} G'$.*

Proof. Suppose that $\xrightarrow[\ell \tau]{=} G = \xrightarrow[\ell \tau]{=} G'$. Then $l \xrightarrow[\ell \tau]{=} G' r$ holds for each equation $l \approx r \in G$. Comparing rules (E4a) and (E4b) from Inference system 2.30 we see that $\xrightarrow[\ell \tau]{=} G' \subseteq \xrightarrow[\ell \tau]{=} G$; hence, we have $l \xrightarrow[\ell \tau]{=} G' r$ for each $l \approx r \in G$. This implies $\xrightarrow[\ell \tau]{=} G \subseteq \xrightarrow[\ell \tau]{=} G'$. By symmetry we obtain $\xrightarrow[\ell \tau]{=} G \subseteq \xrightarrow[\ell \tau]{=} G'$; thus, $\xrightarrow[\ell \tau]{=} G$ and $\xrightarrow[\ell \tau]{=} G'$ are equal. \square

Corollary 4.18. *Every overloaded completion procedure may be used as a nonoverloaded completion procedure. If a set of equations can be completed using an overloaded completion procedure, then its $\mathcal{C}\mathcal{F}$ -semantics and its $\mathcal{N}\mathcal{C}\mathcal{F}$ -semantics agree.*

Proof. Suppose that an overloaded completion procedure is applied to a set G_0 of equations and a set R_0 of sort-decreasing and reductive rules and returns the complete rewrite system R_∞ . Since

$$\xrightarrow[\ell \tau]{=} G_0 \cup R_0 \cup E = \xrightarrow[\ell \tau]{=} R_\infty \cup E = (\xrightarrow{*}_{E \setminus R_\infty} \circ \xleftarrow{*}_{E^\circ} \xleftarrow{*}_{E \setminus R_\infty}),$$

we have by Theorem 4.17 and Corollary 3.11

$$\overline{\overline{\mathcal{L}\mathcal{Y}}} G_0 \cup R_0 \cup E = \overline{\overline{\mathcal{L}\mathcal{Y}}} R_1 \cup E = (\overset{*}{\rightarrow}_{E \setminus R_2} \circ \overset{*}{\leftarrow}_{E \circ} \overset{*}{\leftarrow}_{E \setminus R_2});$$

thus, $\overline{\overline{\mathcal{L}\mathcal{Y}}} G_0 \cup R_0 \cup E$ equals $\overline{\overline{\mathcal{L}\mathcal{Y}}} G_0 \cup R_0 \cup E$. \square

However, there is no corresponding theorem for nonoverloaded completion algorithms. In the following example we show that there are specifications that can be completed in the nonoverloaded, but not in the overloaded world.

Example 4.19. Let Σ be defined by

$$(\mathcal{S}, \leq) = \{s_3 \leq s_1 \leq s_0, s_3 \leq s_2 \leq s_0\},$$

$$\Sigma = \{a: \rightarrow s_1, b: \rightarrow s_2, c: \rightarrow s_3, f: s_1 \rightarrow s_0, f: s_2 \rightarrow s_0, f: s_3 \rightarrow s_0\}.$$

Consider the set of equations $G = \{a \approx b, a \approx f(a), c \approx f(b)\}$ and the complete term rewrite system $R = \{a \rightarrow c, b \rightarrow c, f(c) \rightarrow c\}$. It is easy to prove that $l \overline{\overline{\mathcal{L}\mathcal{Y}}} R r$ holds for all $l \approx r \in G$ and that $l \overline{\overline{\mathcal{L}\mathcal{Y}}} G r$ holds for all $l \rightarrow r \in R$; thus, $\overline{\overline{\mathcal{L}\mathcal{Y}}} G = \overline{\overline{\mathcal{L}\mathcal{Y}}} R$. If we replace inference rule (C2) by

$$(C2)' \text{ Adding an equation } \frac{G, R}{G \cup \{l \approx r\}, R} \quad \text{if } l \overline{\overline{\mathcal{L}\mathcal{Y}}} G \cup R \cup E r$$

(which is correct only for the $\mathcal{N}\mathcal{C}\mathcal{Y}$ -case and, of course, totally impractical), we can, in fact, complete G and obtain R .

Now consider the following $(\Sigma, G)\text{-}\mathcal{L}\mathcal{Y}$ -algebra A :

$$A_{s_0} := \{\alpha, \alpha', \alpha''\}, \quad A_{s_1} := \{\alpha, \alpha'\}, \quad A_{s_2} := \{\alpha, \alpha'\}, \quad A_{s_3} := \{\alpha\},$$

$$A_a^{e, s_1} := \alpha', \quad A_b^{e, s_2} := \alpha', \quad A_c^{e, s_3} := \alpha,$$

$$A_f^{s_1, s_0} : \alpha \mapsto \alpha'', \alpha' \mapsto \alpha', \quad A_f^{s_2, s_0} : \alpha \mapsto \alpha'', \alpha' \mapsto \alpha, \quad A_f^{s_3, s_0} : \alpha \mapsto \alpha''.$$

The algebra A does not satisfy the equation $a \approx c$; hence, $\overline{\overline{\mathcal{L}\mathcal{Y}}} G \neq \overline{\overline{\mathcal{L}\mathcal{Y}}} R$.

Assume that there is a complete term rewrite system R' inducing the same equality as G . Since a and b and G -equal, there must be some term t such that $a \overset{*}{\xrightarrow{\mathcal{L}\mathcal{Y}}} R' t \overset{*}{\xleftarrow{\mathcal{L}\mathcal{Y}}} R' b$. The signature Σ is regular and we have $\text{LS}(a) = s_1$ and $\text{LS}(b) = s_2$; as R' is sort-decreasing, $\text{spectr}(t)$ must contain both s_1 and s_2 , i.e. $\text{LS}(t) = s_3$. Obviously, t cannot be a variable; so, t must be equal to c , which is the only nonvariable term having the least sort s_3 . This, however, is impossible because a and c are not $G\text{-}\mathcal{L}\mathcal{Y}$ -equal. Thus, G cannot be completed using an overloaded completion procedure.

4.5. Remarks

The completion procedure that has been presented in this section differs from an unsorted algorithm in only two places.

In the first place there is a difference with respect to the computation of critical pairs. In general, unification in an order-sorted signature is much less efficient than in the unsorted case. The restriction to regular signatures (so that unification is at least finitary) does not pose many problems; however, even in regular signatures μ CSUs may become quite large, producing a considerable number of critical pairs. Besides, the unifiability problem for regular signatures is NP-complete [39]. Conditions for unitary unifiability [31, 41] are unfortunately seldom satisfied in practical cases.

The second difference is the demand for sort-decreasingness. It would be advantageous if we could content ourselves with the compatibility of the rewrite rules since it is easy to make a signature compatible by construction [39], but compatibility is not sufficient to prove Theorem 4.16 (it is no longer possible to eliminate variable overlaps). Sort-decreasingness, however, constitutes a considerable restriction for many applications. Moreover, the problem to decide whether a rewrite rule is not sort-decreasing is also NP complete, even for signatures with only two sorts $s \leq s'$. (This can be proved by reduction of the MONOTONE 3SAT problem [12, 23] to non-sort-decreasingness.)

5. Conclusion

We have demonstrated which restrictions are necessary for order-sorted equational logic, rewriting, unification and completion. In particular, we have shown that overloaded and nonoverloaded semantics sometimes differ substantially. Whereas conditions such as strong consistency and coherence are necessary for overloaded equational logic, in the nonoverloaded world we can get along without any additional prerequisites. (The well-known problem that terms like $pop(pop(push(push(s, x), y)))$ may be semantically meaningful but syntactically ill-formed can usually be avoided by using error supersorts [14, 39].)

Some severe restrictions, however, become necessary for completion, i.e. for the transformation of an equational specification into an efficient decision procedure.

The translation of order-sorted specifications into conditional many-sorted specifications that Ganzinger proposed [11] might be a way to overcome these problems. Here it is sometimes possible to create complete systems containing non-sort-decreasing rules, but unfortunately the translation considerably increases the size of a specification. Perhaps the advantages of the purely order-sorted and of the conditional many-sorted method could be combined using a partial translation.

Several authors have proposed avoiding the disadvantages of order-sorted rewriting by using another kind of typing [7, 30, 44]. The order-sorted logics that we have considered in this paper have a syntactic sort theory. Using syntactic sorts, the

sorts are not closed under equality, i.e. two terms may be equal with respect to some equational theory E , but nevertheless have different, even disjoint, spectra. In a semantic sort theory there is an axiom like $t : s, t \approx t' \Rightarrow t' : s$. This eliminates the need for compatibility or sort-decreasingness and also the problem of semantically meaningful but syntactically ill-formed terms, but raises a new problem. It is not longer decidable, whether a term t has a sort s , whether a term is an instance of another term, or whether a rule or equation may be applied to a term.

A third possibility, proposed by Comon [5], is to use an unsorted calculus (without imposing a well-typedness condition) with containment predicates $t \in s$ as constraints. A great advantage of this method is that deciding the solvability of such a constraint is often much easier than actually solving it and that the expensive computation of minimal complete sets of order-sorted unifiers can often be avoided.

A comparison of the expressiveness and computational power of these extensions might be an interesting topic for future research.

Acknowledgment

I am grateful to Harald Ganzinger for finding a number of bugs in a previous version of this paper, and to a TCS referee for his valuable detailed comments. I thank Axel Poigné, Gert Smolka and all members of the Lehrstuhl Informatik V for their support.

References

- [1] L. Bachmair, Proof methods for equational theories, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Department of Computer Science, 1987.
- [2] L. Bachmair, Proof methods for equational theories, SUNY at Stony Brook, Department of Computer Science, 1988.
- [3] A.G. Cohn, On the solution of Schubert's Steamroller in many sorted logic, in: A. Joshi, ed., *Proc. 9th Internat. Joint Conf. Artificial Intelligence*, Los Angeles (1985) 1169–1174.
- [4] H. Comon, Unification et disunification: Théorie et applications, Thèse de doctorat, Institut National Polytechnique de Grenoble, 1988.
- [5] H. Comon, Equational formulas in order-sorted algebras, in: M.S. Paterson, ed., *Automata, Languages and Programming, 17th Internat. Colloquium*, Warwick, England, Lecture Notes in Computer Science, Vol. 443 (Springer, Berlin, 1990) 674–688.
- [6] N. Dershowitz and J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B*, Chapter 6 (Elsevier, Amsterdam, 1990) 243–320.
- [7] L. Duponcheel, Typed completion (extended abstract), in: S. Kaplan and M. Okada, eds., *The Second Internat. Workshop on Conditional and Typed Rewrite Systems*, Montreal, extended abstracts (CEN-PARMI, Concordia University, Montreal, 1990) 73–80.
- [8] H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics* (Springer, Berlin, 1985).
- [9] K. Futatsugi, J.A. Goguen, J.-P. Jouannaud and J. Meseguer, Principles of OBJ-2, in: B. Reid, ed., *Proc. 12th ACM Symposium on Principles of Programming Languages* (ACM, New York, 1985) 52–66.

- [10] J.H. Gallier and T. Isakowitz. Rewriting in order-sorted equational logic, in: R.A. Kowalski and K.A. Bowen, eds., *Logic Programming. Proceedings of the Fifth International Conference and Symposium*, Vol. 1 (MIT Press, Cambridge, MA, 1988) 280–294.
- [11] H. Ganzinger. Order-sorted completion: The many-sorted way, Forschungsbericht 274, Universität Dortmund, Fachbereich Informatik, 1988; revised version *Theoret. Comput. Sci.* **89** (1991) 3–32.
- [12] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, San Francisco, 1979).
- [13] I. Gnaedig, C. Kirchner and H. Kirchner. Equational completion in order-sorted algebras (extended abstract), in: M. Dauchet and M. Nivat, eds., *13th Colloquium on Trees in Algebra and Programming*, Nancy, Lecture Notes in Computer Science, Vol. 299 (Springer, Berlin, 1988) 165–184; revised version in: *Theoret. Comput. Sci.* **72** (1990) 169–202.
- [14] M. Gogolla. Algebraic specifications with partially ordered sorts and declarations, Forschungsbericht 169, Universität Dortmund, Abteilung Informatik, 1983; revised version: Partially ordered sorts in algebraic specifications, in: B. Courcelle, ed., *Proc. Ninth Colloquium on Trees in Algebra and Programming* (Cambridge University Press, Cambridge, 1984) 139–153.
- [15] M. Gogolla, Über partiell geordnete Sortenmengen und deren Anwendung zur Fehlerbehandlung in abstrakten Datentypen, Dissertation, Universität Braunschweig, Naturwissenschaftliche Fakultät, 1986.
- [16] J.A. Goguen, J.-P. Jouannaud and J. Meseguer. Operational semantics for order-sorted algebra, in: W. Brauer, ed., *12th Internat. Coll. on Automata, Languages and Programming*, Nafplion, Greece, Lecture Notes in Computer Science, Vol. 194 (Springer, Berlin, 1985) 221–231.
- [17] J.A. Goguen and J. Meseguer. Completeness of many-sorted equational logic, *ACM Sigplan Notices* **16** (1981) 24–32.
- [18] J.A. Goguen and J. Meseguer. Order-sorted algebra I: Partial and overloaded operators, errors and inheritance, Draft, SRI International and Center for the Study of Language and Information, 1985.
- [19] J.A. Goguen and J. Meseguer. Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems, in: *Proc. Second Symp. on Logic in Computer Science* (IEEE Computer Society Press, Silver Spring, MD, 1987) 18–29.
- [20] J.A. Goguen and J. Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. Draft, Programming Research Group, University of Oxford and SRI International, 1989.
- [21] J.A. Goguen and J.J. Tardo. An introduction to OBJ, in: M. Zelkowitz, ed., *Specification of Reliable Software* (IEEE Press, New York, 1979) 170–189.
- [22] J.A. Goguen, J.W. Thatcher, F.G. Wagner and J.B. Wright. Initial algebra semantics and continuous algebras, *J. ACM* **24** (1977) 68–95.
- [23] E.M. Gold. Complexity of automaton identification from given data, Unpublished manuscript, cited in [12], 1974.
- [24] J.-P. Jouannaud. Confluent and coherent equational term rewriting systems, applications to proofs in abstract data types, in: G. Ausiello and M. Protasi, eds., *8th Coll. on Trees in Algebra and Programming*, Lecture Notes in Computer Science, Vol. 159 (Springer, Berlin, 1983) 269–283.
- [25] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations, *SIAM J. Comput.* **15** (1986) 1155–1194.
- [26] C. Kirchner. Order-sorted equational unification, Rapport de Recherche 954, Institut National de Recherche en Informatique et en Automatique, 1988.
- [27] C. Kirchner, H. Kirchner and J. Meseguer. Operational semantics of OBJ-3 (extended abstract), in: T. Lepistö and A. Salomaa, eds., *15th Internat. Coll. on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 317 (Springer, Berlin, 1988) 287–301.
- [28] D.E. Knuth and P. Bendix. Simple word problems in universal algebras, in: J. Lecch, ed., *Computational Problems in Abstract Algebra* (Pergamon Press, Oxford, 1970) 263–297.
- [29] H.-J. Kreowski and Z. Qian. Relation-sorted algebraic specifications with built-in coercers: Basic notions and results, in: C. Choffrut and T. Lengauer, eds., *7th Ann. Symp. on Theoretical Aspects of Computer Science*, Rouen, Lecture Notes in Computer Science, Vol. 415 (Springer, Berlin, 1990) 165–175.
- [30] Aristide [Megrelis]. A logic of semi-functions, inclusion and equality. The setting, Tech. Report CRIN 89-R-058, Centre de Recherche en Informatique de Nancy, 1989.

- [31] J. Meseguer, J.A. Goguen and G. Smolka, Order-sorted unification, Tech. Report CSLI-87-86, Center for the Study of Language and Information, 1987; revised version in: *J. Symbolic Comput.* **8** (1989) 383–413.
- [32] T. Nipkow and G. Snelting, Type classes and overloading resolution via order-sorted unification, Tech. Report 200, University of Cambridge, Computer Laboratory, 1990.
- [33] A. Oberschelp, Untersuchungen zur mehrsortigen Quantorenlogik, *Mathematische Annalen* **145** (1962) 297–333.
- [34] A. Poigné, Parametrization for order-sorted algebraic specification, *J. Comput. System Sci.* **40** (1986) 229–268.
- [35] M. Schmidt-Schauß, A many-sorted calculus with polymorphic functions based on resolution and paramodulation, in: A. Joshi, ed., *Proc. 9th Internat. Joint Conf. Artificial Intelligence*, Los Angeles (1985) 1162–1168.
- [36] M. Schmidt-Schauß, Computational aspects of an order-sorted logic with term declarations, Dissertation, Universität Kaiserslautern, Fachbereich Informatik, 1988.
- [37] G. Smolka, Order-sorted Horn logic: Semantics and deduction, SEKI-Report SR-86–17, Universität Kaiserslautern, Fachbereich Informatik, 1986.
- [38] G. Smolka, TEL (Version 0.9): Report and user manual, SEKI-Report SR-87–11, Universität Kaiserslautern, Fachbereich Informatik, 1988.
- [39] G. Smolka, W. Nutt, J.A. Goguen and J. Meseguer, Order-sorted equational computation, SEKI-Report SR-87–14, Universität Kaiserslautern, Fachbereich Informatik, 1987; revised version in: H. Ait-Kaci and M. Nivat, eds., *Resolution of Equations in Algebraic Structures, Vol. 2* (Academic Press, San Diego, CA, 1989) 297–367.
- [40] J.G. Stell, Order sorted theories as monads: A term rewriting viewpoint (extended abstract), in: S. Kaplan and M. Okada, eds., *The Second Internat. Workshop on Conditional and Typed Rewrite Systems*, Montreal, extended abstracts (CENPARMI, Concordia University, Montreal, 1990) 208–215.
- [41] U. Waldmann, Unitary unification in order-sorted signatures (extended abstract), in: H.-J. Bürckert and W. Nutt, eds., *Extended Abstracts of the 3rd Internat. Workshop on Unification*, Lambrecht, FRG (Universität Kaiserslautern, Fachbereich Informatik, SEKI Report SR-89–17, 1989) 106–110.
- [42] U. Waldmann, Compatibility of order-sorted rewrite rules, in: S. Kaplan and M. Okada, eds., *The Second Internat. Workshop on Conditional and Typed Rewrite Systems*, Montreal, extended abstracts (CENPARMI, Concordia University, Montreal, 1990) 198–207.
- [43] C. Walther, A many-sorted calculus based on resolution and paramodulation, in: *Proc. of the 8th Internat. Joint Conf. Artificial Intelligence*, Karlsruhe (1983) 882–891.
- [44] P. Watson and J. Dick, Least sorts in order-sorted term rewriting, Tech. Report CSD-TR-606, University of London, Royal Holloway and Bedford New College, 1989.